Motion Modules in Logix5000 Control Systems

User Manual

Catalog Numbers 1756-HYD02, 1756-L60M03SE, 1756-M02AE, 1756-M02AS, 1756-M03SE, 1756-M08SE, 1756-M16SE, 1768-M04SE

ALLEN-BRADLEY • ROCKWELL SOFTWARE

Rockwell Automation

Important User Information

Solid state equipment has operational characteristics differing from those of electromechanical equipment. Safety Guidelines for the Application, Installation and Maintenance of Solid State Controls (publication SGI-1.1 available from your local Rockwell Automation sales office or online at http://literature.rockwellautomation.com) describes some important differences between solid state equipment and hard-wired electromechanical devices. Because of this difference, and also because of the wide variety of uses for solid state equipment, all persons responsible for applying this equipment must satisfy themselves that each intended application of this equipment is acceptable.

In no event will Rockwell Automation, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Rockwell Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams.

No patent liability is assumed by Rockwell Automation, Inc. with respect to use of information, circuits, equipment, or software described in this manual.

Reproduction of the contents of this manual, in whole or in part, without written permission of Rockwell Automation, Inc., is prohibited.

Throughout this manual, when necessary, we use notes to make you aware of safety considerations.

WARNING	Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.
IMPORTANT	Identifies information that is critical for successful application and understanding of the product.
ATTENTION	Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you identify a hazard, avoid a hazard, and recognize the consequence
SHOCK HAZARD	Labels may be on or inside the equipment, for example, a drive or motor, to alert people that dangerous voltage may be present.
BURN HAZARD	Labels may be on or inside the equipment, for example, a drive or motor, to alert people that surfaces may reach dangerous temperatures.

Allen-Bradley, Rockwell Automation, and TechConnect are trademarks of Rockwell Automation, Inc.

Trademarks not belonging to Rockwell Automation are property of their respective companies.

Introduction

This publication has new and updated information. To find new and updated information, look for change bars, as shown next to this paragraph.

Updated Information

This document has these changes.

Change	See
Updated screen graphics to coincide with software release.	Chapter 4 and Chapter 6
Updated Kinematics chapter to include information regarding Delta two-dimensional, Delta three-dimensional, SCARA Independent, and SCARA Delta information.	Chapter 6

Notes:

Preface

Introduction 9
Description of the Modules 9
Additional Resources 10
Help for Selecting Drives and Motors 10
Where to Find Sample Projects 10

Chapter 1

Start

Introduction 13
Make the Controller the Master Clock 14
Add the Motion Modules 15
Add SERCOS interface Drives 16
Set Up Each SERCOS Interface Module 17
Add the Motion Group 18
Add Your Axes 20
Set Up Each Axis 22
Check the Wiring of Each Drive 25
Tune Each Axis 26
Get Axis Information 27
Program Motion Control 28
What's Next? 30

Chapter 2

Test an Axis with Motion Direct Commands

Introduction 31
Access Motion Direct Commands 32
Choose a Command 34
Motion Direct Command Dialog 37
Motion Direct Command Error Process 39
What If the Software Goes Offline or the Controller Changes Modes? 42
Can Two Workstations Give Motion Direct Commands? 42

Chapter 3

Handle Faults

Introduction 43
Choose If Motion Faults Shut Down the Controller 44
Choose the Fault Actions for an Axis 45
Set the Fault Action for an Axis 46

Chapter 4

Create and Configure a Coordinate System

Introduction 47
Create a Coordinate System 48
Enter Tag Information 48
Coordinate System Wizard Dialogs 50
Edit Coordinate System Properties 51

Chapter 5

Inhibit an Axis

Introduction 69
When to Inhibit an Axis. 69
Before You Begin. 70
Example: Inhibit an Axis 73
Example: Uninhibit an Axis. 74

Chapter 6

Kinematics in RSLogix 5000 Software

Introduction 75
Controllers that Support Kinematics Functionality 75
Overview of Kinematics Functionality in RSLogix 5000 Software . . . 75
Useful Terms 77
Gather Information about Your Robot 77
Summary of Kinematic Steps 78
Determine the Coordinate System Type. 80
Configure an Articulated Independent Robot 82
Configure an Articulated Dependent Robot. 92
Configure a Cartesian Gantry Robot. 101
Configure a Cartesian H-bot 102
Configure a SCARA Independent 104
Configure Delta Robot Geometries 108
Configure a SCARA Delta Robot 122
Arm Solutions. 127
Solution Mirroring for Three-dimensional Robots. 127
Activating Kinematics. 128
Change the Robot Arm Solution. 129
Plan for Singularity. 130
Encounter a No-solution Position 130
Error Conditions 131

Chapter 7

Interpret Module Lights (LEDs)

Introduction 133
1756-M02AE Module 133
1756-M02AS Module. 135
1756-HYD02 Module 138
SERCOS interface Module 141

Chapter 8

Troubleshoot Axis Motion

Introduction 143
Why does my axis accelerate when I stop it? 143
Why does my axis overshoot its target speed? 145
Why is there a delay when I stop and then restart a jog? 148
Why does my axis reverse direction when I stop and start it? 150

Chapter 9

Configure Homing

Introduction 153
Guidelines for Homing 153
Examples 154

Appendix A

Axis Properties

Introduction 159
General Tab – AXIS_SERVO 159
General Tab - AXIS_SERVO_DRIVE 160
General Tab - AXIS_VIRTUAL. 164
General Tab – AXIS_GENERIC. 165
Motion Planner Tab. 166
Units Tab 169
Servo Tab - AXIS_SERVO. 170
Feedback Tab – (AXIS_SERVO) 172
Drive/Motor Tab - (AXIS_SERVO_DRIVE) 177
Motor Feedback Tab - AXIS_SERVO_DRIVE 184
Aux Feedback Tab - AXIS_SERVO_DRIVE. 185
Conversion Tab 187
Homing Tab - AXIS_SERVO. 188
Homing Tab - AXIS_SERVO_DRIVE. 193
Homing Tab - AXIS_VIRTUAL 197
Hookup Tab - AXIS_SERVO 198
Hookup Tab Overview - AXIS_SERVO_DRIVE 200
Tune Tab - AXIS_SERVO, AXIS_SERVO_DRIVE. 202
Dynamics Tab - AXIS_SERVO, AXIS_SERVO _DRIVE, AXIS_VIRTUAL. 205
Gains Tab - AXIS_SERVO. 210
Gains Tab - AXIS_SERVO_DRIVE. 215
Output Tab - AXIS_SERVO 222
Output Tab Overview - AXIS_SERVO_DRIVE 225
Limits Tab - AXIS_SERVO 229
Limits Tab - AXIS_SERVO_DRIVE. 233
Offset Tab - AXIS_SERVO 239
Offset Tab - AXIS_SERVO_DRIVE 242
Fault Actions Tab - AXIS_SERVO 246
Fault Actions Tab - AXIS_SERVO_DRIVE. 249
Tag Tab. 254

Appendix B

Wiring Diagrams

Introduction 257
1756-M02AE Module 258
Ultra 100 Series Drive 259
Ultra 200 Series Drive 259
Ultra3000 Drive 261
1394 Servo Drive (in Torque Mode only) 263
1756-M02AS Module 265
1756-HYD02 Application Example 266
1756-HYD02 Module 267
LDTs 268
Temposonic GH Feedback Device 269
24V Registration Sensor 270
5V Registration Sensor 270
Home Limit Switch Input 271
OK Contacts 271

Appendix C

Axis Attributes

Introduction 273
How to Access Attributes 273
Axis Attributes 274
Additional Error Code Information 384

Appendix D

Servo Loop Block Diagrams

Introduction 387
Interpreting the Diagrams 387
AXIS_SERVO 388
AXIS_SERVO_DRIVE 390

Appendix E

Axis Data Types

Introduction 399
AXIS_CONSUMED 399
AXIS_GENERIC 402
AXIS_SERVO 404
AXIS_SERVO_DRIVE 407
AXIS_VIRTUAL 411

Appendix F

Coordinate System Attributes

How to Access Attributes 413
Coordinate System Attributes 414

Introduction

Use this manual to setup and program motion control using these Logix5000 motion modules.

Description of the Modules

This table describes the Logix5000 motion modules.

Motion Module	Description
1756-M02AE	The 1756-M02AE is a two-axis servo module for drives/actuators that need a ±10V velocity or torque reference. Use the 1756-M02AE when your equipment has quadrature encoder feedback. The module also has: · Home limit switch inputs · Drive fault inputs · Drive enable outputs · 5V or 24V position registration inputs · 250 µs position and velocity loop updates
1756-HYD02	The 1756-HYD02 is a two-axis servo module for hydraulic actuators that need a ±10V velocity reference. Use the 1756-HYD02 when your equipment has magnostrictive linear transducer (LDT) feedback. The module is similar to the 1756-M02AE with these exceptions: · Feed Forward adjust in addition to single-step Auto Tune. · Gain ratio between extend direction and retract direction to accommodate hydraulic cylinder dynamics. · Intelligent transducer noise detection filtering in hardware and firmware replaces programmable IIR filtering.
1756-M02AS	The 1756-M02AS is a two-axis servo module for drives/actuators that need a ±10 volt velocity or torque reference input. Use the 1756-M02AS when your equipment has Serial Synchronous Input (SSI) position feedback. The module is similar to the 1756-M02AE with these exceptions: · Gain ratio between extend direction and retract direction to accommodate hydraulic cylinder dynamics. · Intelligent transducer noise detection filtering in hardware and firmware replaces programmable IIR filtering. · SSI interface consisting of Differential Clock output and Data return signals replaces the differential encoder interface.
1756-M03SE 1756-M08SE 1756-M16SE 1768-M04SE	Use a SERCOS interface module to connect the controller to SERCOS interface drives. · The SERCOS interface lets you control digital drives using high-speed, real time, serial communication. · SERCOS is the IEC 61491 SErial Real-time Communication System protocol over a fiber optic network. · The module uses a fiber optic network for all the wiring between the drives and the module.

Additional Resources

See these manuals for more information about using motion modules in a Logix5000 control system.

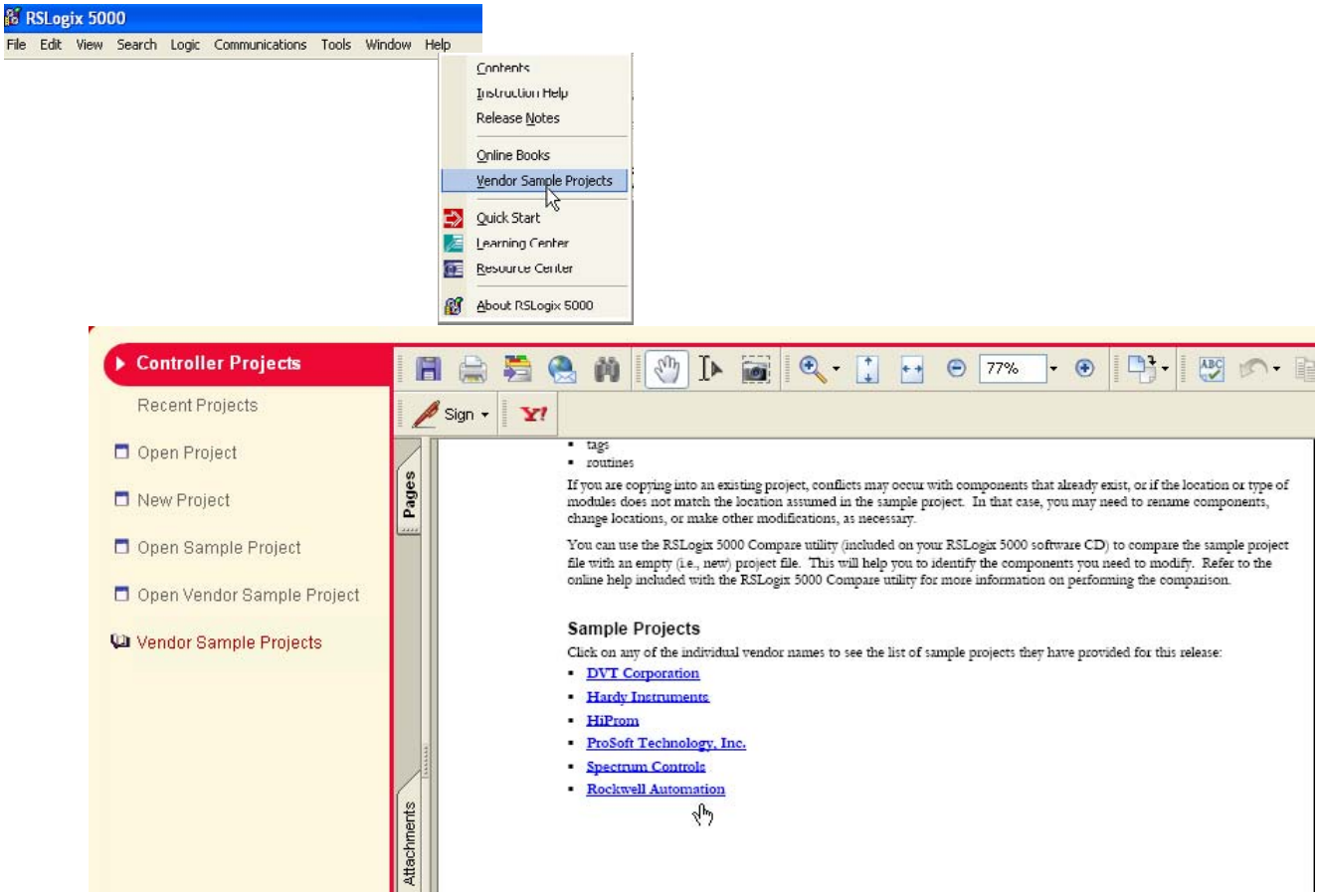
Publication	Publication Number
Logix5000 Controllers Quick Start	1756-QS001
Logix5000 Controllers Common Procedures	1756-PM001
Logix5000 Controller Motion Instructions Reference Manual	1756-RM007
Logix5000 Controllers General Instructions Reference Manual	1756-RM003
Logix5000 Controllers Process and Drives Instructions Reference Manual	1756-RM006
PhaseManager User Manual	LOGIX-UM001
ControlLogix Controller User Manual	1756-UM001
CompactLogix Controllers User Manual	1768-UM001
Analog Encoder (AE) Servo Module Installation Instructions	1756-IN047
ControlLogix SERCOS interface Module Installation Instructions	1756-IN572
CompactLogix SERCOS interface Module Installation Instructions	1768-IN005
1394 SERCOS Interface Multi Axis Motion Control System Installation Manual	1394-IN002
1394 SERCOS Integration Manual	1394-IN024
Ultra3000 Digital Servo Drives Installation Manual	2098-IN003
Ultra3000 Digital Servo Drives Integration Manual	2098-IN005
Kinetix 6000 Installation Manual	2094-IN001
Kinetix 6000 Integration Manual	2094-IN002
8720MC High Performance Drive Installation Manual	8720MC-IN001
8720MC High Performance Drive Integration Manual	8720MC-IN002

Help for Selecting Drives and Motors

The Motion Analyzer utility helps you select the appropriate Rockwell drives and motors based upon your load characteristics and typical motion application cycles. The Motion Analyzer guides you through wizard-like screens to collect information specific to your application. After you enter the information (such as load inertia, gear box ratio, feedback device and brake requirements all available through the robot manufacturer) the Motion Analyzer generates an easy-to-read list of recommended motors, drives and other support equipment to interface with the type of robot you are using.

Where to Find Sample Projects

Sample projects from Rockwell Automation as well as other vendors are available from the RSLogix 5000 software Help system menu.


RSLogix 5000
File Edit View Search Logic Communications Tools Window Help
Contents
Instruction Help
Release Notes
Online Books
Vendor Sample Projects
Quick Start
Learning Center
Resource Center
About RSLogix 5000
Controller Projects
Recent Projects
Open Project
New Project
Open Sample Project
Open Vendor Sample Project
Vendor Sample Projects
77%
Sign
Pages
Attachments
tags
routines
If you are copying into an existing project, conflicts may occur with components that already exist, or if the location or type of modules does not match the location assumed in the sample project. In that case, you may need to rename components, change locations, or make other modifications, as necessary.
You can use the RSLogix 5000 Compare utility (included on your RSLogix 5000 software CD) to compare the sample project file with an empty (i.e., new) project file. This will help you to identify the components you need to modify. Refer to the online help included with the RSLogix 5000 Compare utility for more information on performing the comparison.
Sample Projects
Click on any of the individual vendor names to see the list of sample projects they have provided for this release:
DVT Corporation
Hardy Instruments
HiProm
ProSoft Technology, Inc.
Spectrum Controls
Rockwell Automation

Notes:

Start

Introduction

Use this chapter for step-by-step procedures on how to set up motion control.

IMPORTANT If you aren't using SERCOS interface drives and modules, skip tasks 3 and 4.

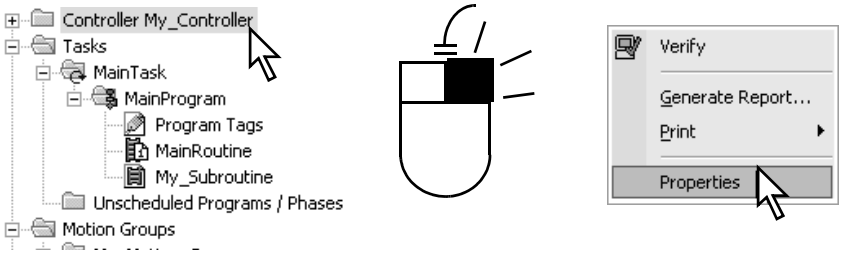
Topic	See page
1. Make the Controller the Master Clock	14
2. Add the Motion Modules	15
3. Add SERCOS interface Drives	16
4. Set Up Each SERCOS Interface Module	17
5. Add the Motion Group	18
6. Add Your Axes	20
7. Set Up Each Axis	22
8. Check the Wiring of Each Drive	25
9. Tune Each Axis	26
10. Get Axis Information	27
11. Program Motion Control	28
12. What's Next?	30

Make the Controller the Master Clock

You must make one module in the chassis the master clock for motion control. This module is called the coordinated system time (CST) master. The motion modules set their clocks to the CST master.

In most cases, make the controller the CST master.

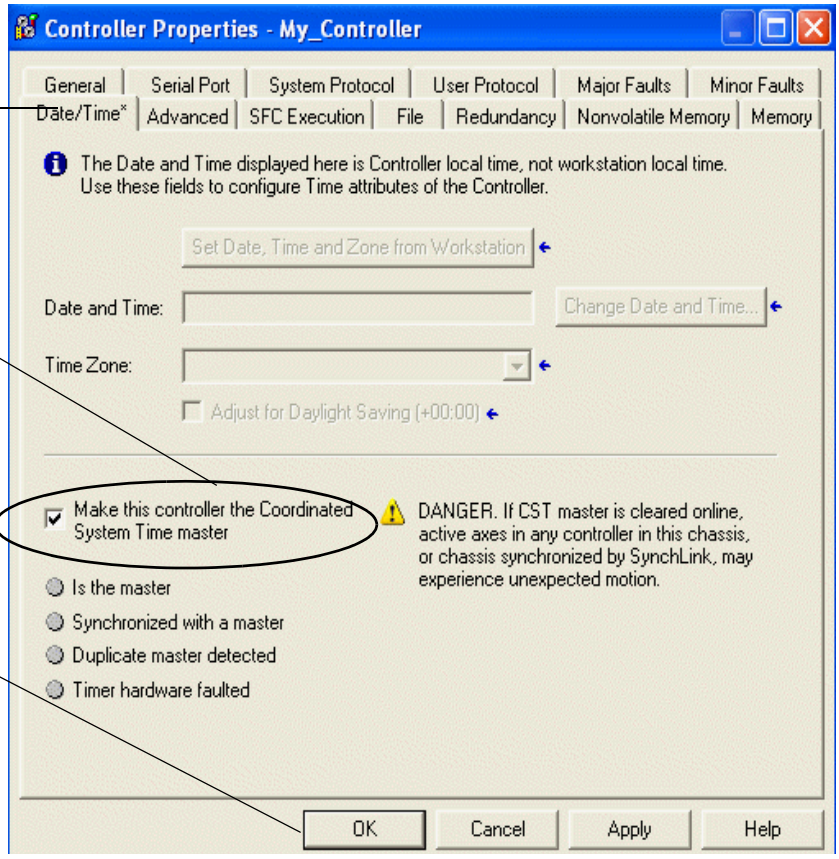
1.



2.

3.

4.



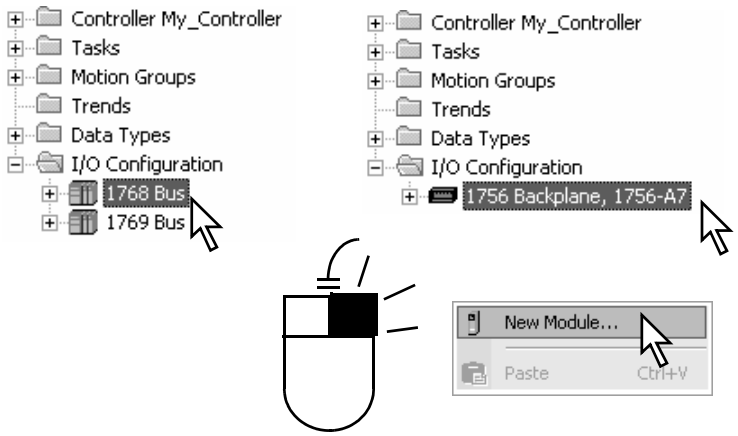
If you have more than one controller in the chassis

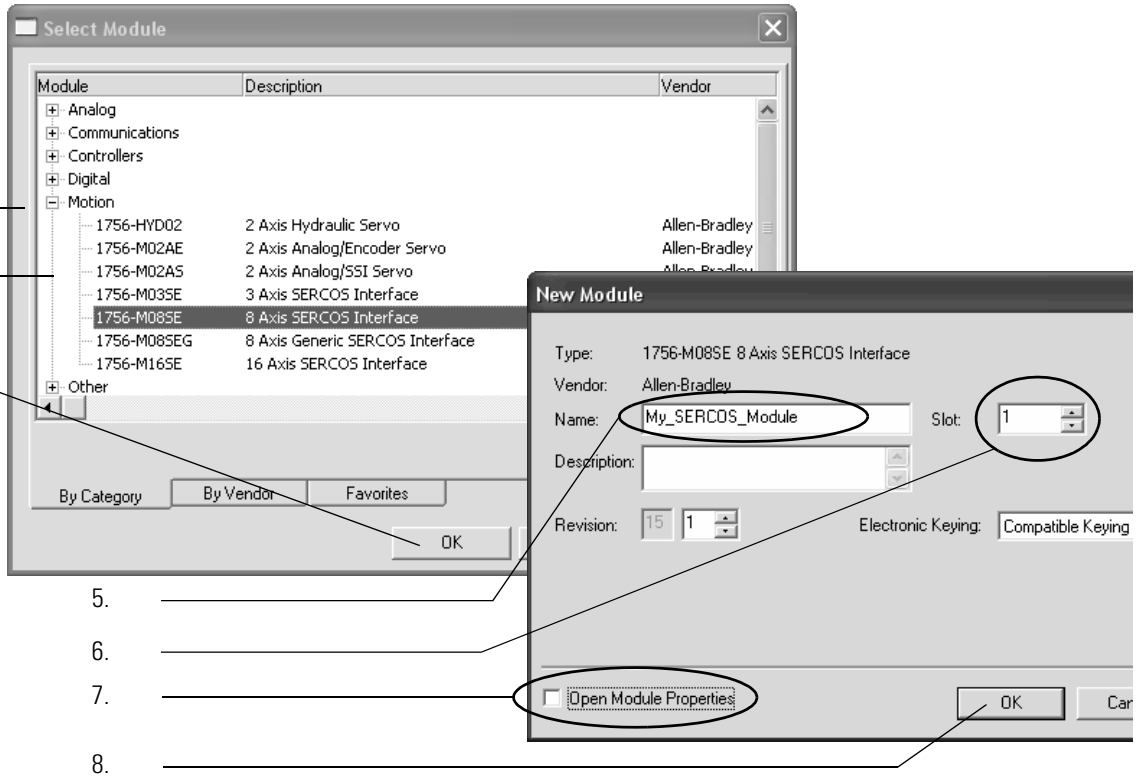
If you have more than one controller in the chassis, choose one of the controllers to be the CST master. You can't have more than one CST master for the chassis.

Add the Motion Modules

IMPORTANT For your motion modules, use the firmware revision that goes with the firmware revision of your controller. See the release notes for your controller's firmware.

1. **CompactLogix controller** **ControlLogix controller**

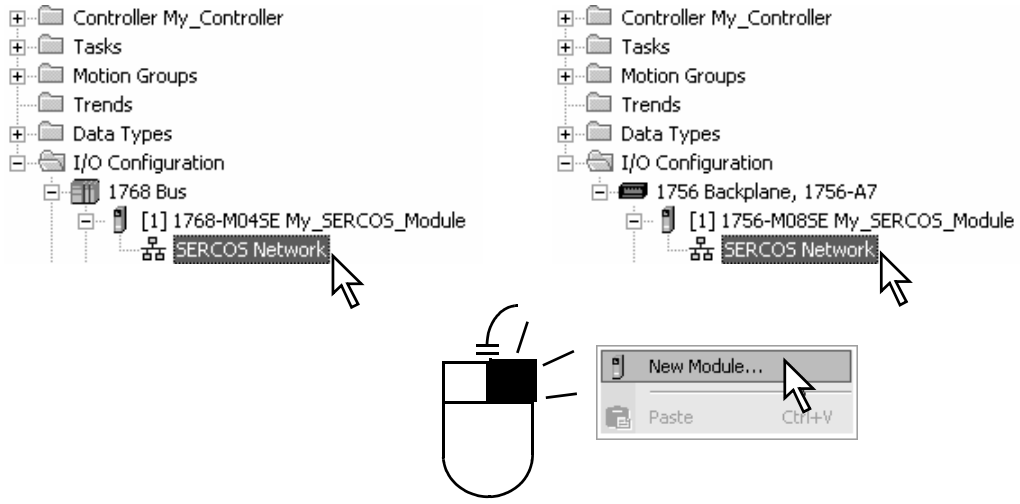


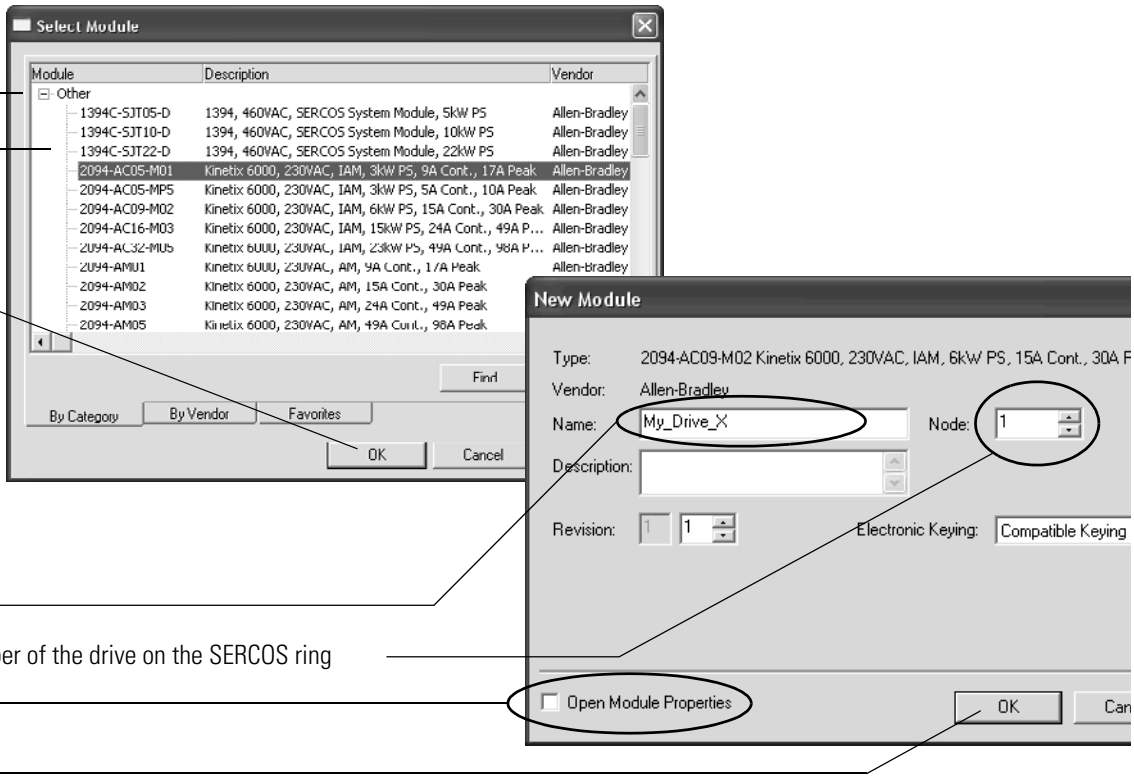


Add SERCOS interface Drives

Add SERCOS interface drives to the I/O configuration of the controller. This lets you use RSLogix 5000 software to set up the drives.

1. **CompactLogix controller** **ControlLogix controller**



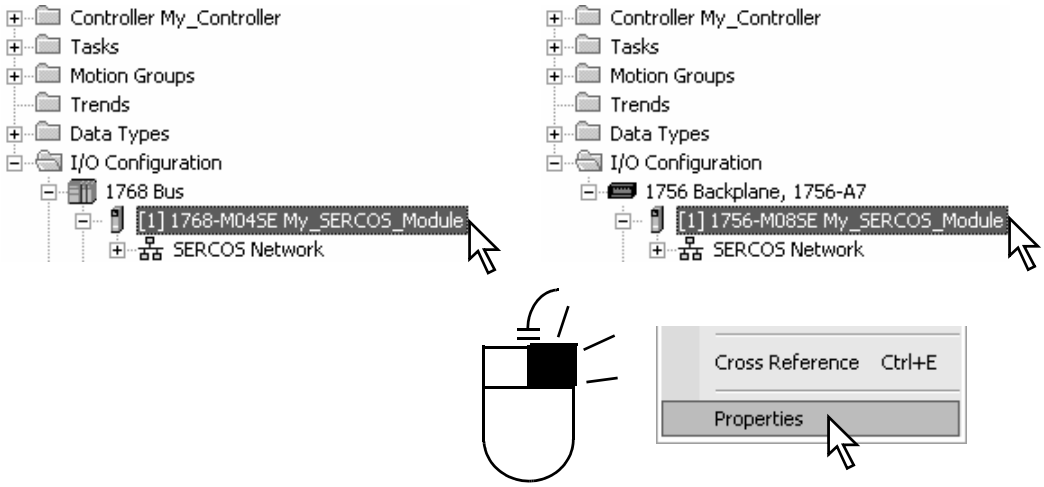

Select Module
Module
Description
Vendor
Other
1394C-SJT05-D 1394, 460VAC, SERCOS System Module, 5kW PS Allen-Bradley
1394C-SJT10-D 1394, 460VAC, SERCOS System Module, 10kW PS Allen-Bradley
1394C-SJT22-D 1394, 460VAC, SERCOS System Module, 22kW PS Allen-Bradley
2094-AC05-M01 Kinetix 6000, 230VAC, IAM, 3kW PS, 9A Cont., 17A Peak Allen-Bradley
2094-AC05-MP5 Kinetix 6000, 230VAC, IAM, 3kW PS, 5A Cont., 10A Peak Allen-Bradley
2094-AC09-M02 Kinetix 6000, 230VAC, IAM, 6kW PS, 15A Cont., 30A Peak Allen-Bradley
2094-AC16-M03 Kinetix 6000, 230VAC, IAM, 15kW PS, 24A Cont., 49A P... Allen-Bradley
2094-AC32-M05 Kinetix 6000, 230VAC, IAM, 23kW PS, 49A Cont., 98A P... Allen-Bradley
2094-AM01 Kinetix 6000, 230VAC, AM, 9A Cont., 17A Peak Allen-Bradley
2094-AM02 Kinetix 6000, 230VAC, AM, 15A Cont., 30A Peak
2094-AM03 Kinetix 6000, 230VAC, AM, 24A Cont., 49A Peak
2094-AM05 Kinetix 6000, 230VAC, AM, 49A Cont., 98A Peak
Find
By Category
By Vendor
Favorites
OK
Cancel
New Module
Type: 2094-AC09-M02 Kinetix 6000, 230VAC, IAM, 6kW PS, 15A Cont., 30A P
Vendor: Allen-Bradley
Name: My_Drive_X
Node: 1
Description:
Revision: 1 1
Electronic Keying: Compatible Keying
Open Module Properties
OK

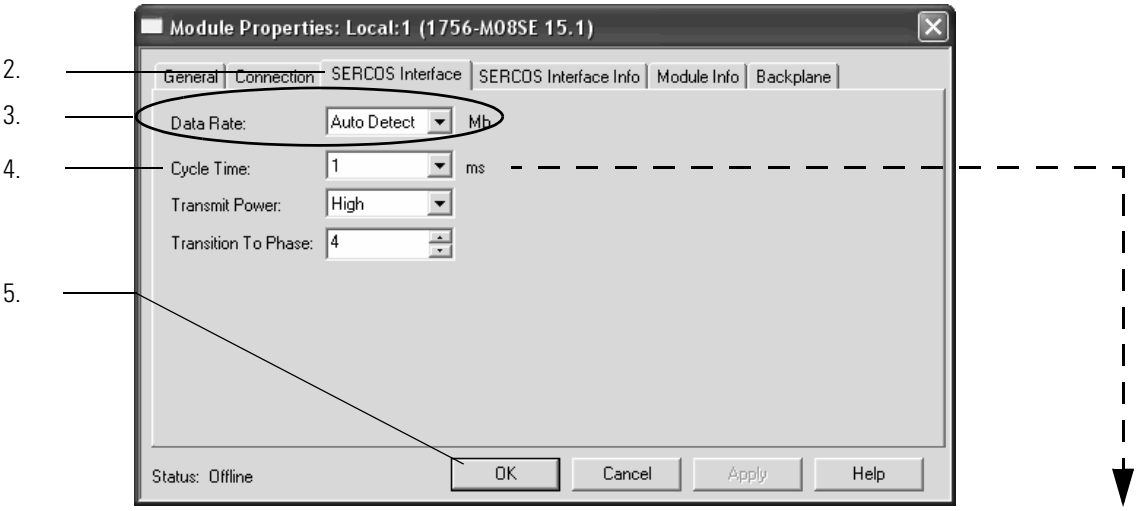
2. _____
3. _____
4. _____
5. _____
6. Node number of the drive on the SERCOS ring
7. _____
8. _____

Set Up Each SERCOS Interface Module

Set the data rate and cycle time for each SERCOS interface module in your project.

1. **CompactLogix controller** **ControlLogix controller**



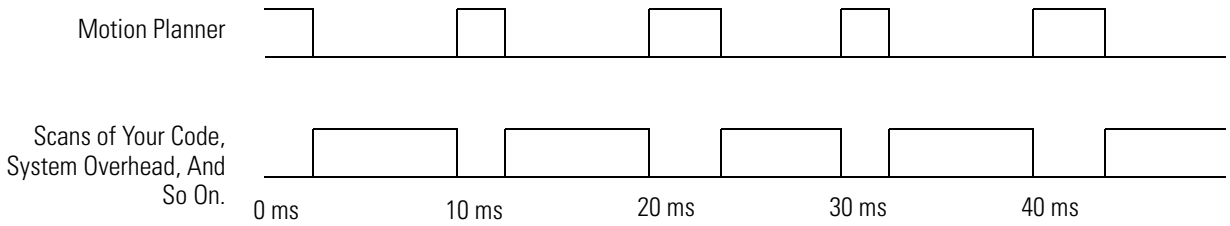


Baud Rate of Drives	Number of Drives on the Ring	Type of Drives	Cycle Time
4 Mb	1 or 2	Kinetix 6000	0.5 ms
		NOT Kinetix 6000	1 ms
	3 or 4	→	1 ms
	5…8	→	2 ms
	9…16	→	Can't do.
8 Mb	1…4	Kinetix 6000	0.5 ms
		NOT Kinetix 6000	1 ms
	5…8	→	1 ms
	9…16	→	2 ms

Add the Motion Group

Add a motion group to set up the motion planner.

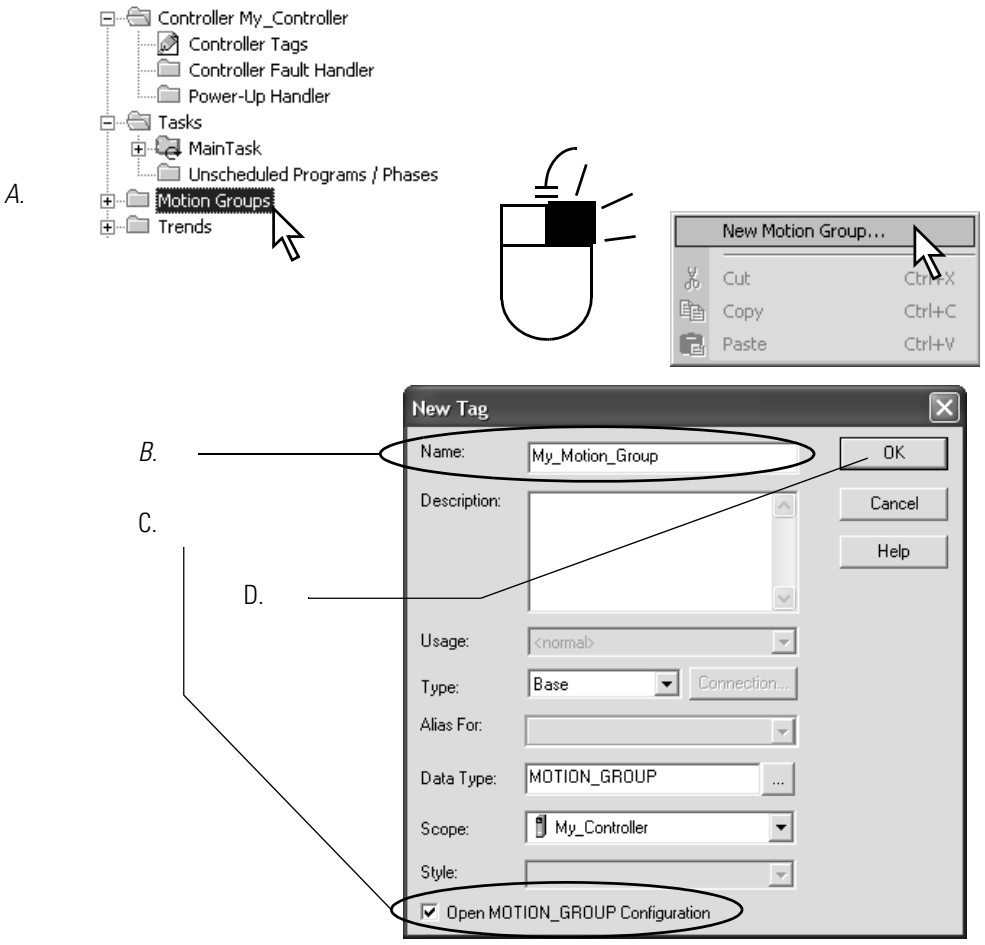
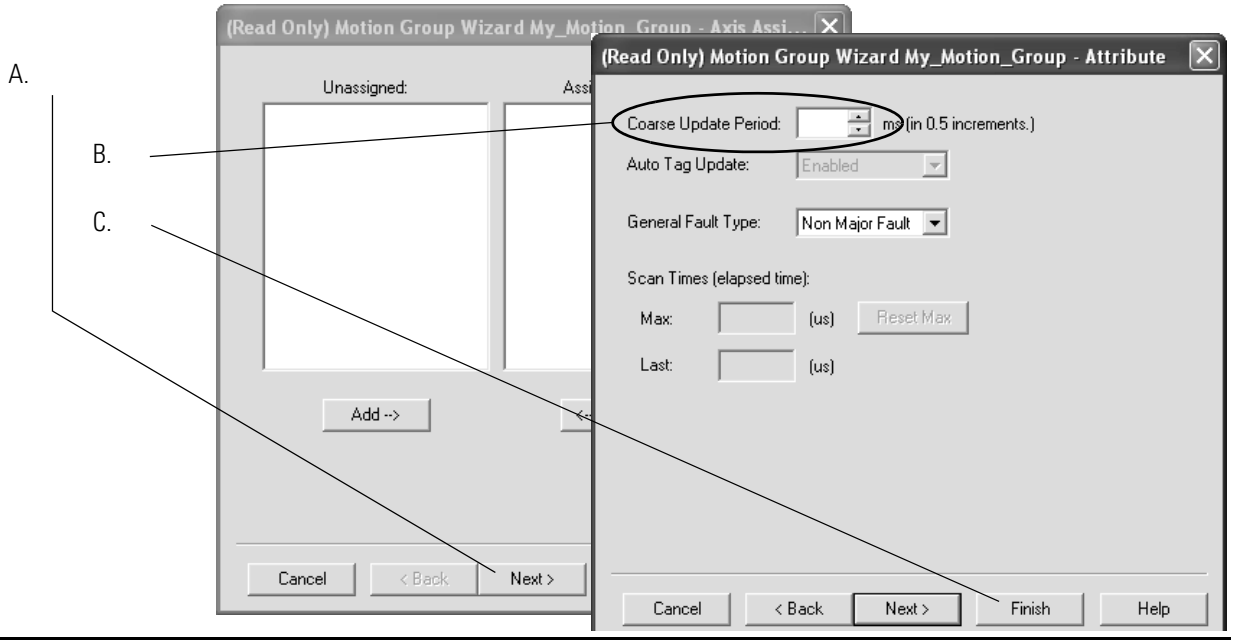
Motion Planner	Part of the controller that takes care of position and velocity information for your axes
Coarse Update Period	How often the motion planner runs. When the motion planner runs, it interrupts all other tasks regardless of their priority.



In this example, the coarse update period = 10 ms. Every 10 ms the controller stops scanning your code and whatever else it is doing and runs the motion planner.

IMPORTANT Add only one motion group for the project. RSLogix 5000 software doesn't let you add more than one motion group.

Action	Details
1. Choose your coarse update period.	The coarse update period is a trade-off between updating positions of your axes and scanning your code. Use these guidelines as a rough starting point. A. How many axes do you have? · Less than 11 axes — Set the coarse update period to 10 ms. · 11 axes or more — Set the coarse update period to 1 ms per axis. B. Leave at least half the controller's time for the scan of all your code. C. If you have SERCOS interface motion modules, set the coarse update period to a multiple of the cycle time of the motion module. Example: if the cycle time is 2 ms, set the coarse update period to 8 ms, 10 ms, 12 ms, and so on. D. If you have analog motion modules, set the coarse update period to: 1. At least 3 times the servo update period of the motion module 2. A multiple of the servo update period of the motion module

Action	Details
2. Add the motion group.	
3. Set the coarse update period.	

Add Your Axes

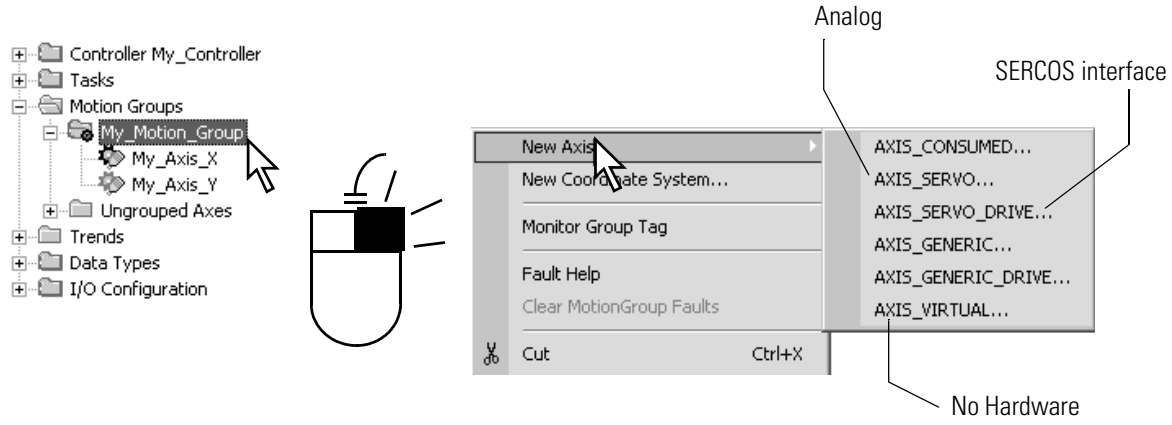
Add an axis for each of your drives.

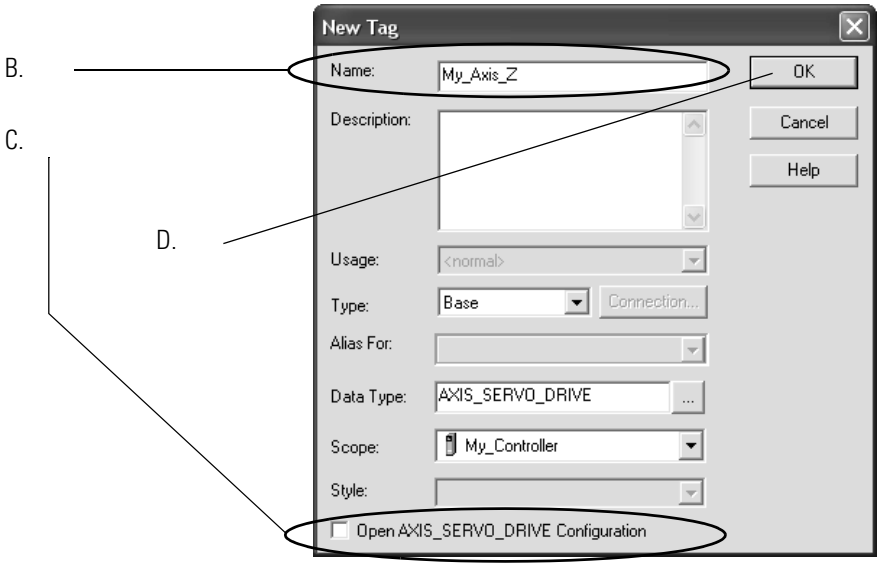
Action	Details	
1. Decide which data type to use.	**If you use this motion module for the axis**	**Then use this data type**
	1756-M03SE	AXIS_SERVO_DRIVE
	1756-M08SE	
	1756-M16SE	
	1756-L60M03SE	
	1768-M04SE	
	1756-M02AE	AXIS_SERVO
	1756-HYD02	
	1756-M02AS	
	No hardware	AXIS_VIRTUAL

Action	Details

2. Add an axis.

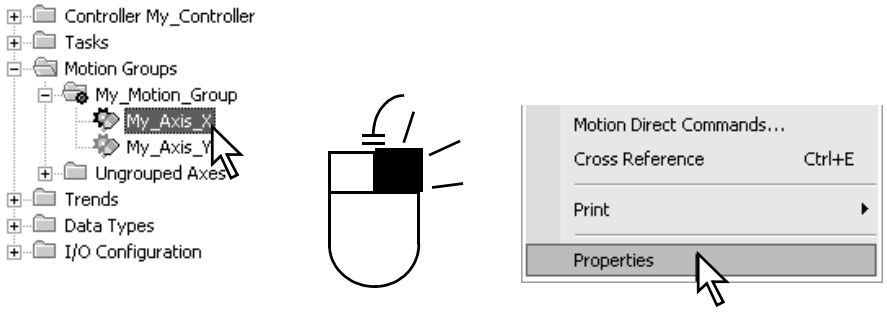
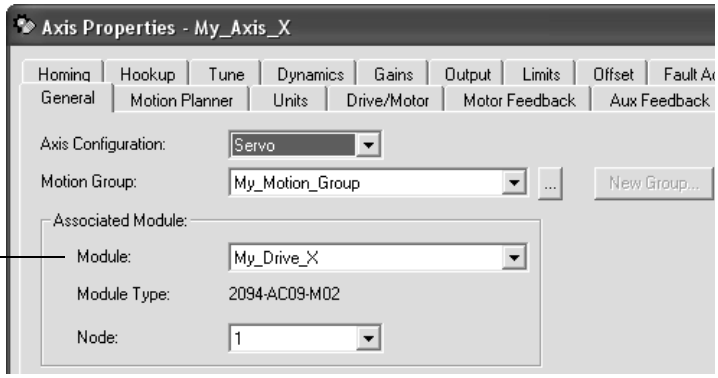
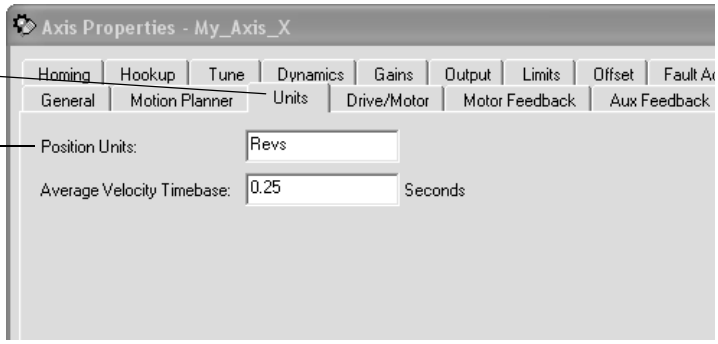
A.

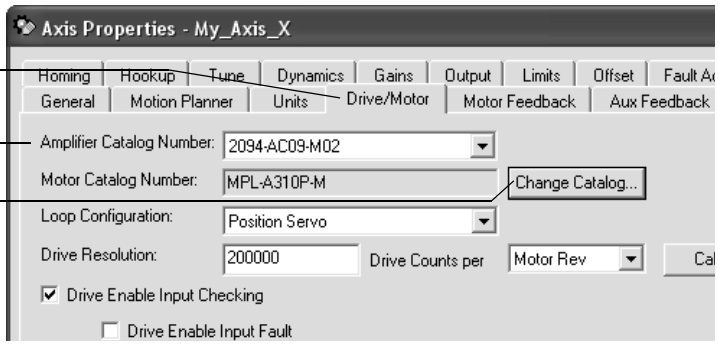
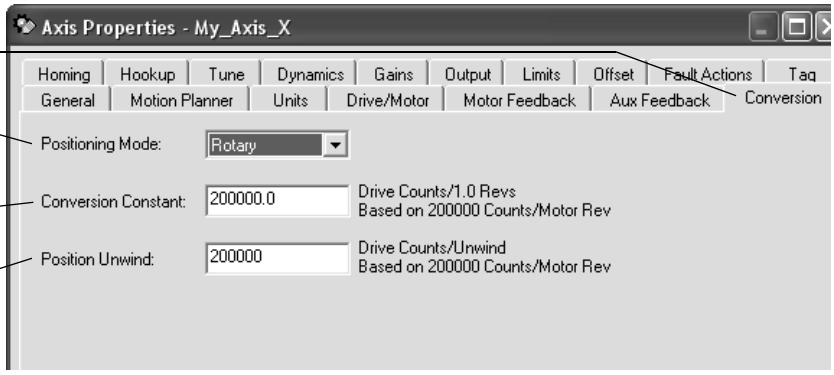
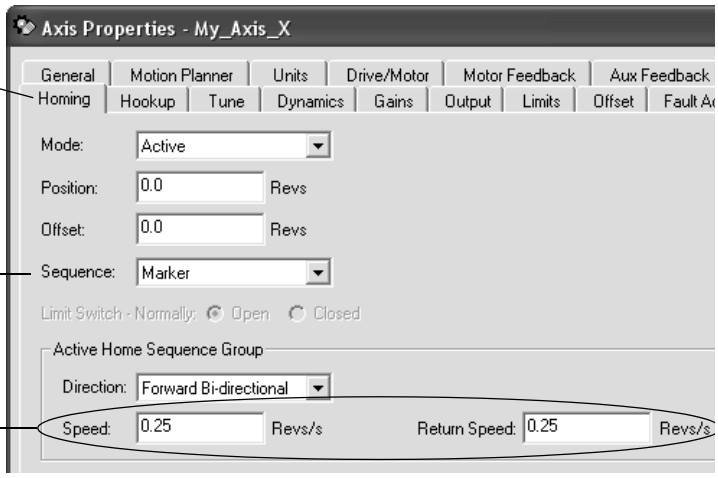


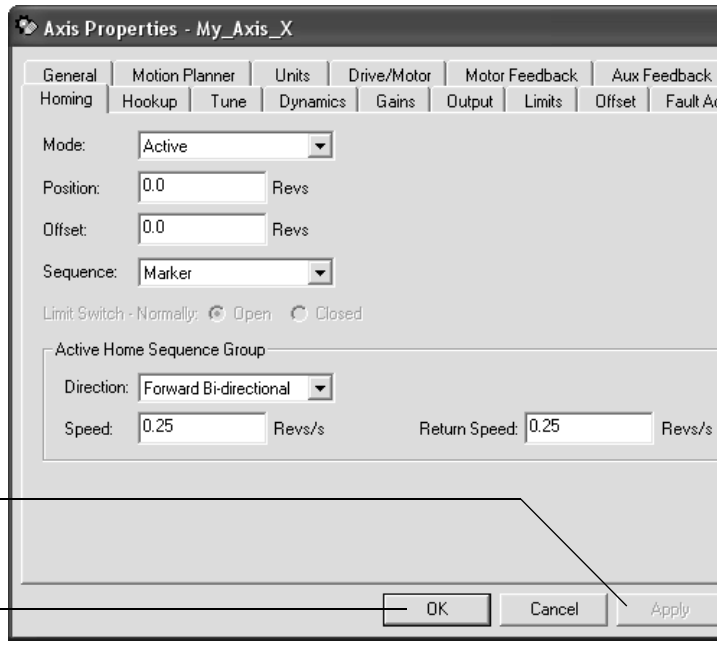


Set Up Each Axis

The following steps show how to set up the axis of a SERCOS interface drive. The steps are slightly different if you have a different type of drive.

Action	Details
1. Open the properties for the axis.	
2. Select the drive for the axis. Select the name that you gave to the drive for this axis.	
3. Set the units that you want to program in. A. B. Type the units that you want to use for programming, such as revs, degrees, inches, or millimeters.	

Action	Details
4. Select the drive and motor catalog numbers. A. B. Select the catalog number of the drive. C. Select the catalog number of the motor.	
5. Set the conversion between drive counts and units. A. B. Select whether this is a rotary or linear axis. C. Type the number of drive counts that equal one unit from step 3B. D. If this is a rotary axis, type the number of drive counts that you want to unwind after.	
6. Set up the homing sequence. A. B. Select the type of homing sequence that you want. C. Type homing speeds.	

Action	Details
7. Apply your changes.	

Check the Wiring of Each Drive

Use the hookup tests to check the wiring of a drive.

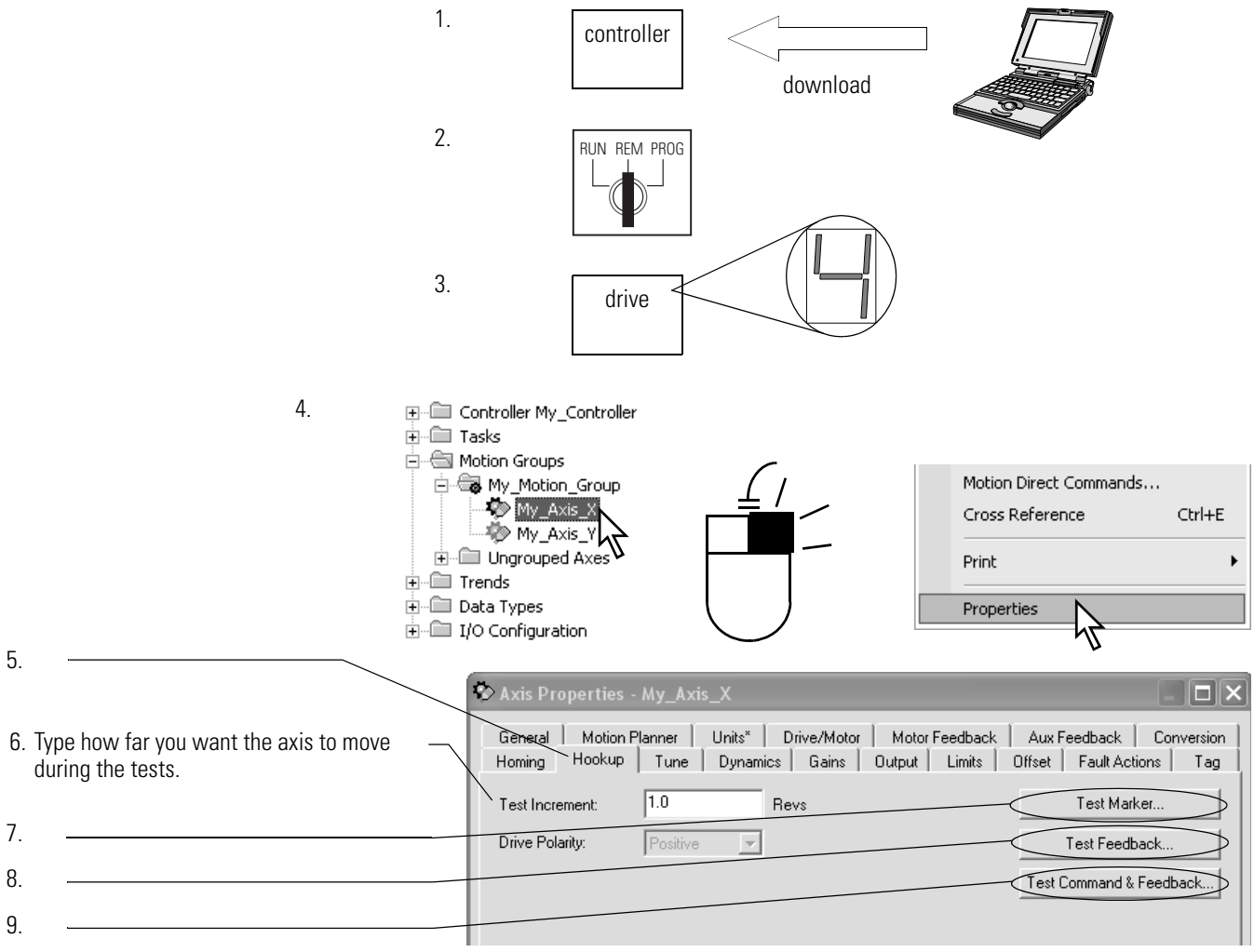
This Test	Does This	Notes
Test marker	Checks that the drive gets the marker pulse.	You must manually move the axis for this test.
Test feedback	Checks the polarity of the feedback.	You must manually move the axis for this test.
Test command and feedback	Checks the polarity of the drive.	

ATTENTION

These tests make the axis move even with the controller in remote program mode.

- Before you do the tests, make sure no one is in the way of the axis.
- **Do not** change the polarity after you do the tests. Otherwise you may cause an axis-runaway condition.

1.
2.
3.
4.
5.
6. Type how far you want the axis to move during the tests.
7.
8.
9.



Tune Each Axis

Use the Tune tab to tune an axis.

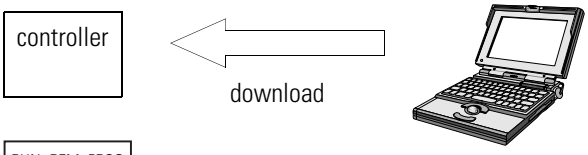
ATTENTION

When you tune an axis, it moves even with the controller in remote program mode. In that mode, your code is ***not*** in control of the axis.

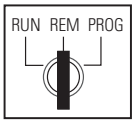
Before you tune an axis, make sure no one is in the way of the axis.

The default tuning procedure tunes the proportional gains. Typically, tune the proportional gains first and see how your equipment runs.

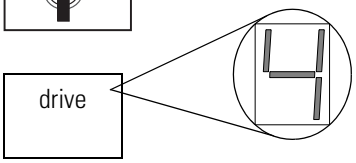
1.



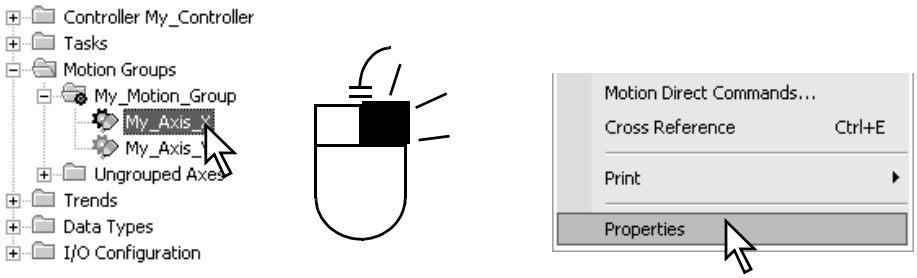
2.



3.



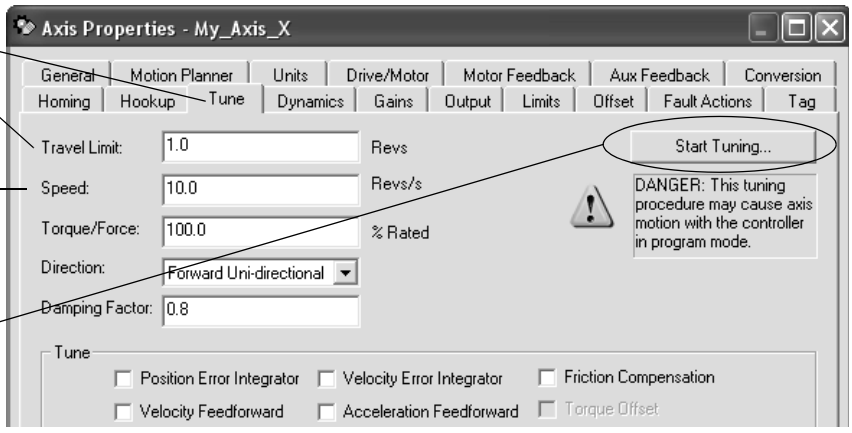
4.



5.

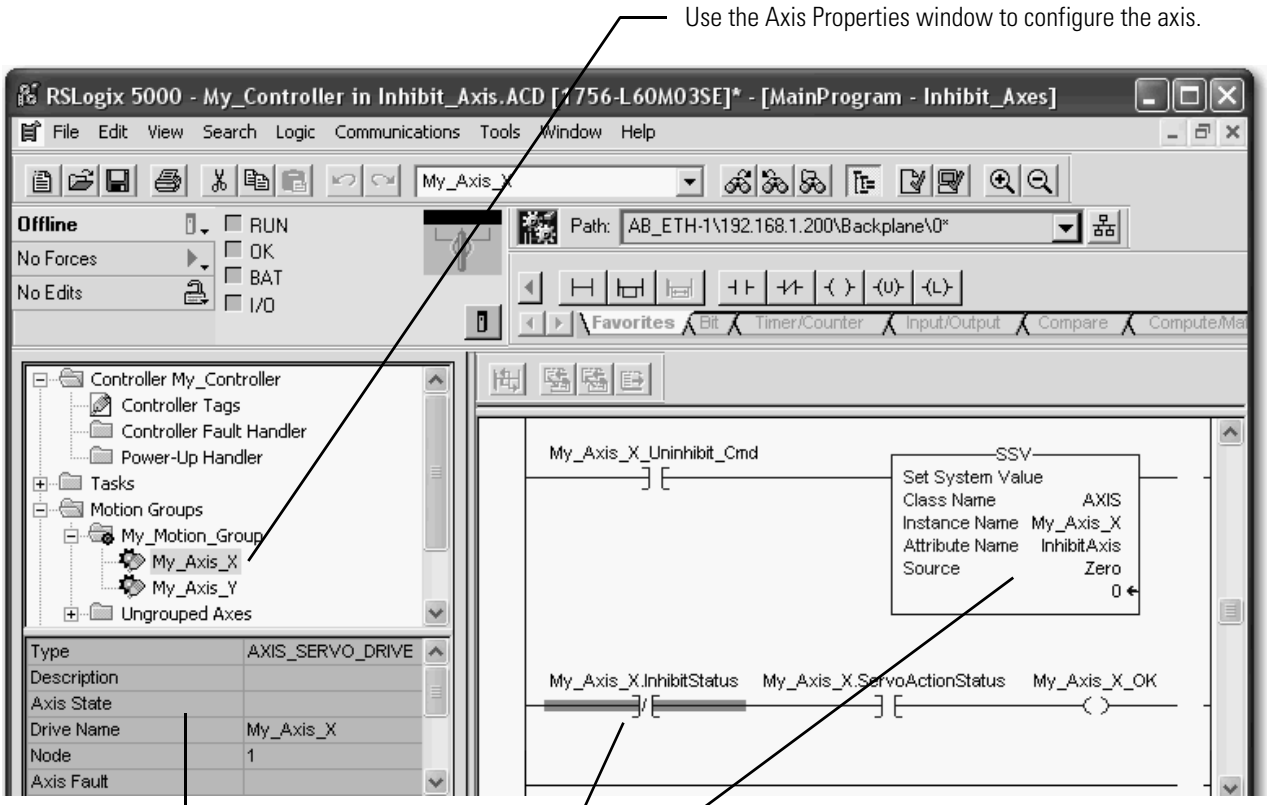
6. Type the limit of movement for the axis during the tuning procedure.

7. Type the maximum speed for your equipment.



Get Axis Information

You can get information about an axis in several ways.



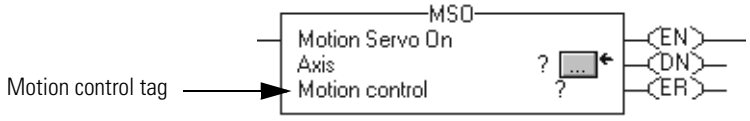
Program Motion Control

See:

- *Logix5000 Controllers Common Procedures Manual*, 1756-PM001
- *Logix5000 Controllers Motion Instructions Reference Manual*, 1756-RM007
- *Logix5000 Controllers General Instructions Reference Manual*, 1756-RM003

The controller gives you a set of motion control instructions for your axes.

- Uses these instructions just like the rest of the Logix5000 instructions. You can program motion control in these programming languages:
  - ladder diagram (LD)
  - structured text (ST)
  - sequential function chart (SFC)
- Each motion instruction works on one or more axes.
- Each motion instruction needs a motion control tag. The tag uses a MOTION_INSTRUCTION data type. The tag stores the status information of the instruction.



ATTENTION

Use the tag for the motion control operand of motion instruction only once. Unintended operation of the control variables may happen if you re-use of the same motion control tag in other instructions.

Example

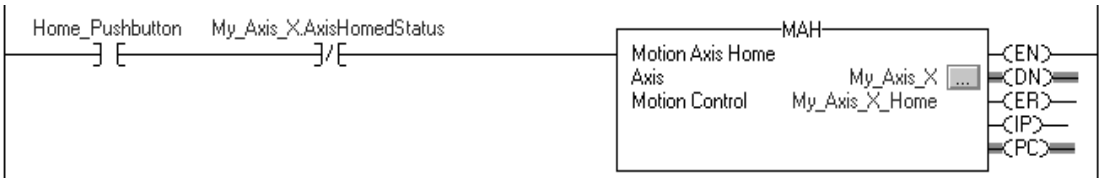
Here's an example of a simple ladder diagram that homes, jogs, and moves an axis.

If *Initialize_Pushbutton* = on and the axis = off (*My_Axis_X.ServoActionStatus* = off) then

The MSO instruction turns on the axis.

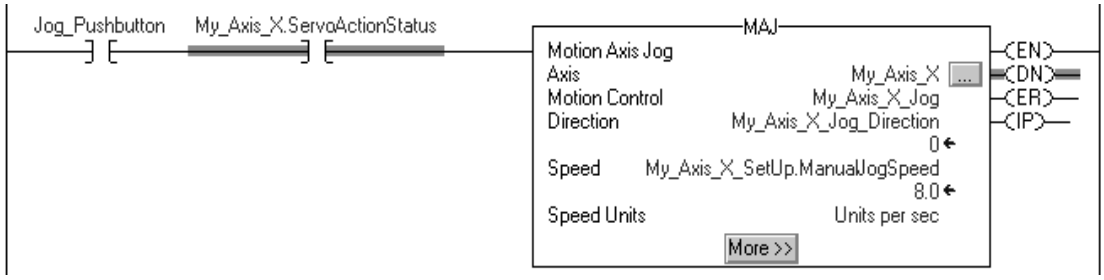
If *Home_Pushbutton* = on and the axis hasn't been homed (*My_Axis_X.AxisHomedStatus* = off) then

The MAH instruction homes the axis.



If *Jog_Pushbutton* = on and the axis = on (*My_Axis_X.ServoActionStatus* = on) then

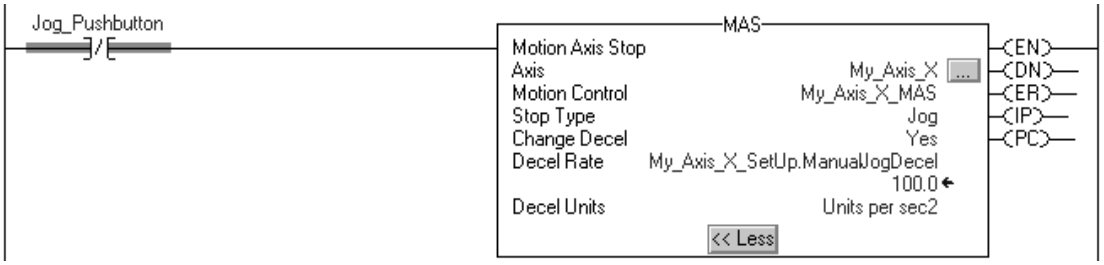
The MAJ instruction jogs the axis forward at 8 units/s.



If *Jog_Pushbutton* = off then

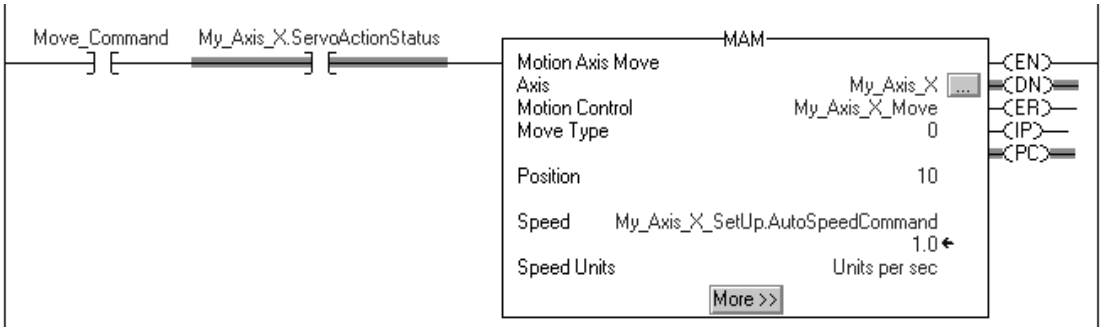
The MAS instruction stops the axis at 100 units/s^2

Make sure that *Change Decel* is *Yes*. Otherwise, the axis decelerates at its maximum speed.



If *Move_Command* = on and the axis = on (*My_Axis_X.ServoActionStatus* = on) then

The MAM instruction moves the axis. The axis moves to the position of 10 units at 1 unit/s.



What's Next?

Use these chapters to continue programming your motion control system.

- Test an Axis with Motion Direct Commands
- Configure Homing
- Handle Faults
- Create and Configure a Coordinate System
- Inhibit an Axis

Test an Axis with Motion Direct Commands

Introduction

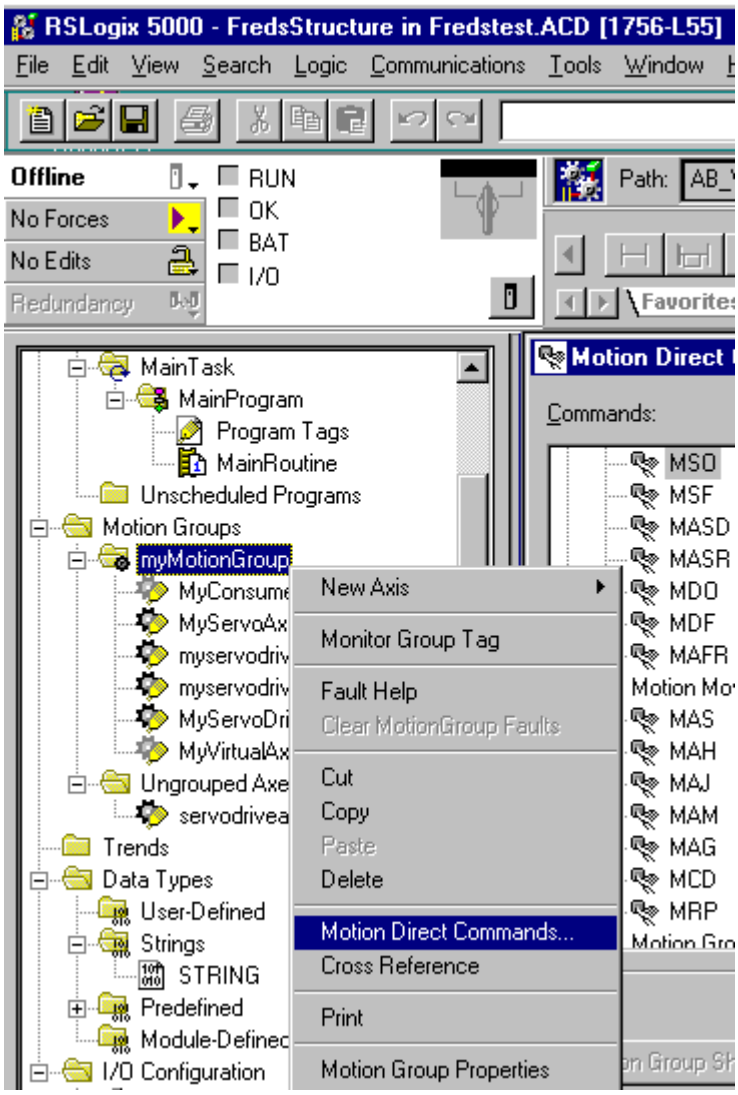
The Motion Direct Commands feature lets you issue motion commands while you are online without having to write or execute an application program. Motion Direct Commands are particularly useful when you are commissioning or debugging a motion application. During commissioning, you can configure an axis and monitor the behavior using Trends in the Controller Organizer. Use of Motion Direct Commands can "fine-tune" the system with or without load to optimize its performance. When in the testing and or debugging cycle, you can issue Motion Direct Commands to establish or reestablish conditions such as Home. Often during initial development or enhancement to mature applications you need to test the system in small manageable areas. These tasks include:

- Home to establish initial conditions
- Incrementally Move to a physical position
- Monitor system dynamics under specific conditions

Access Motion Direct Commands

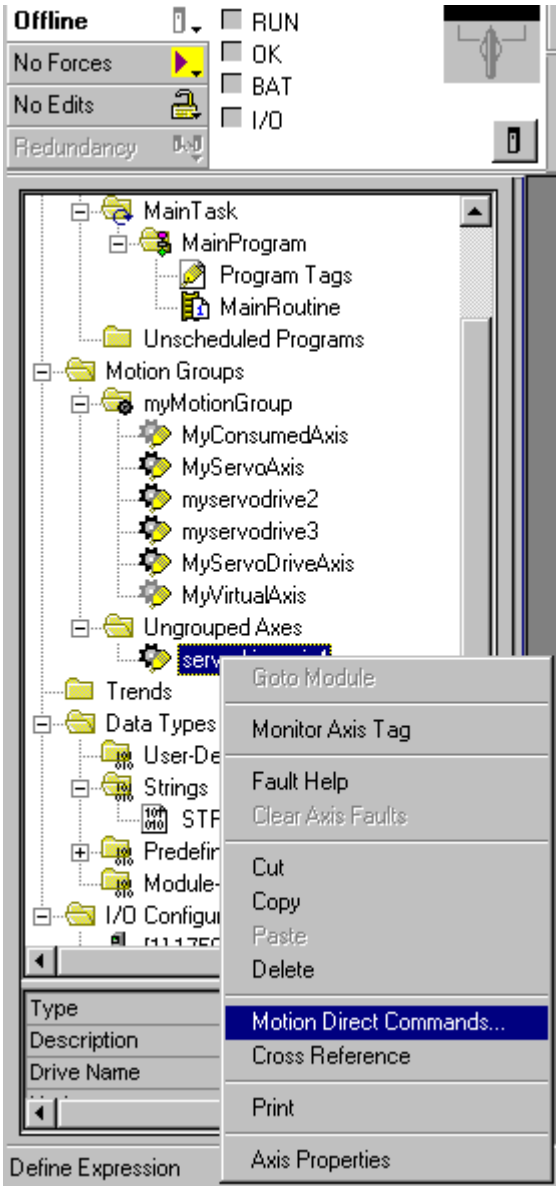
Access the Motion Direct Commands for the Motion Group

To access the Motion Direct Commands for the motion group, right- click the group in the Controller Organizer.



Access the Motion Direct Commands for an Axis

To access the Motion Direct Commands for an axis, right-click the axis in the Controller Organizer.



Choose a Command

Use this table to choose an instruction and see if it is available as a Motion Direct Command.

If you want to	And	Use this instruction	Motion direct Command
Change the state of an axis	Enable the servo drive and activate the axis servo loop.	MSO Motion Servo On	Yes
	Disable the servo drive and deactivate the axis servo loop.	MSF Motion Servo Off	Yes
	Force an axis into the shutdown state and block any instructions that initiate axis motion.	MASD Motion Axis Shutdown	Yes
	Transition an axis to the ready state. If all of the axes of a servo module are removed from the shutdown state as a result of this instruction, the OK relay contacts for the module close.	MASR Motion Axis Shutdown Reset	Yes
	Enable the servo drive and set the servo output voltage of an axis.	MDO Motion Direct Drive On	Yes
	Disable the servo drive and set the servo output voltage to the output offset voltage.	MDF Motion Direct Drive Off	Yes
	Clear all motion faults for an axis.	MAFR Motion Axis Fault Reset	Yes
Control axis position	Stop any motion process on an axis.	MAS Motion Axis Stop	Yes
	Home an axis.	MAH Motion Axis Home	Yes
	Jog an axis.	MAJ Motion Axis Jog	Yes
	Move an axis to a specific position.	MAM Motion Axis Move	Yes
	Start electronic gearing between 2 axes	MAG Motion Axis Gear	Yes
	Change the speed, acceleration, or deceleration of a move or a jog that is in progress.	MCD Motion Change Dynamics	Yes
	Change the command or actual position of an axis.	MRP Motion Redefine Position	Yes
	Calculate a Cam Profile based on an array of cam points.	MCCP Motion Calculate Cam Profile	No
	Start electronic camming between 2 axes.	MAPC Motion Axis Position Cam	No
	Start electronic camming as a function of time.	MATC Motion Axis Time Cam	No
	Calculate the slave value, slope, and derivative of the slope for a cam profile and master value.	MCSV Motion Calculate Slave Values	No

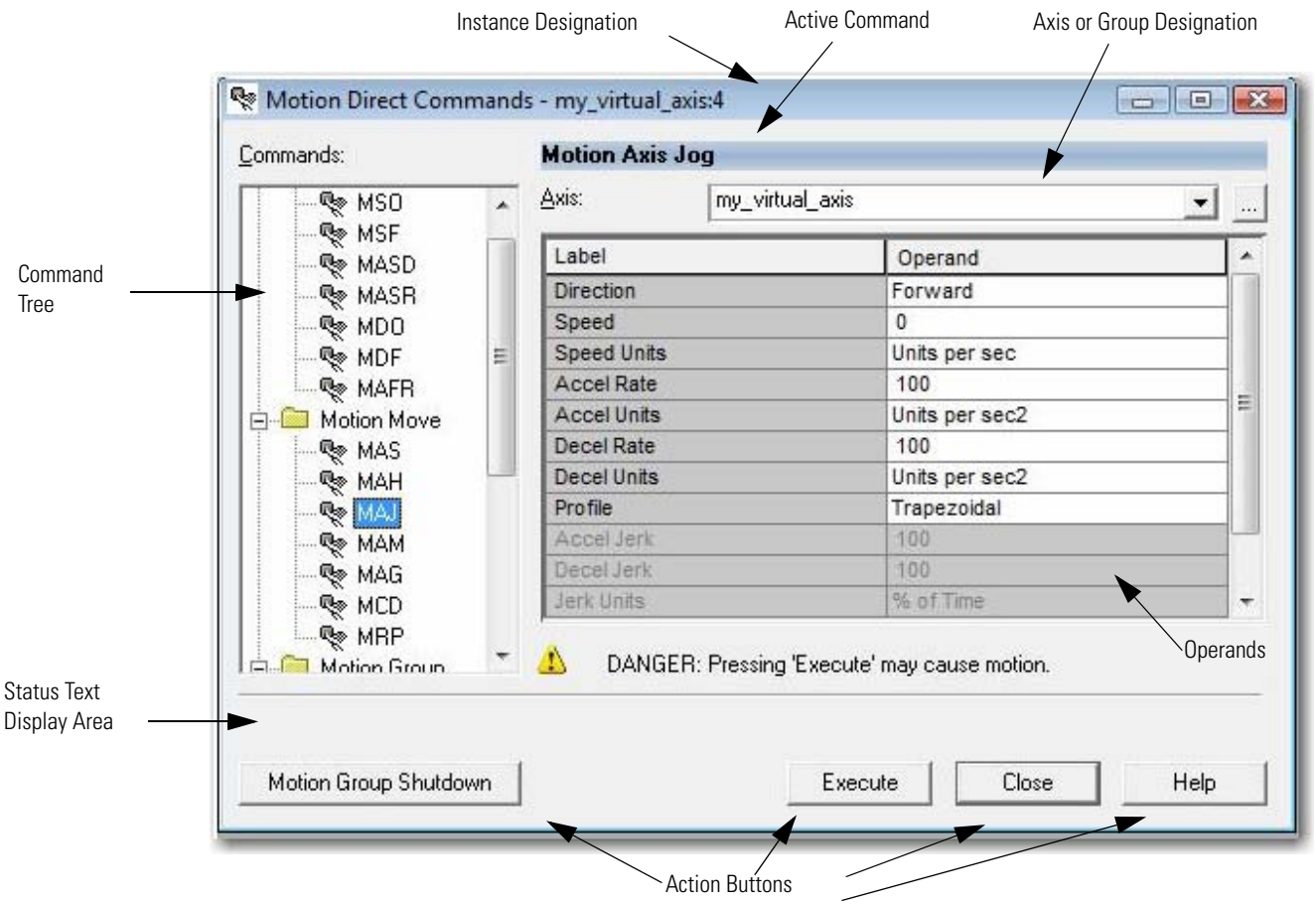
If you want to	And	Use this instruction	Motion direct Command
Initiate action on all axes	Stop motion of all axes.	MGS Motion Group Stop	Yes
	Force all axes into the shutdown state.	MGSD Motion Group Shutdown	Yes
	Transition all axes to the ready state.	MGSR Motion Group Shutdown Reset	Yes
	Latch the current command and actual position of all axes.	MGSP Motion Group Strobe Position	Yes
Arm and disarm special event checking functions such as registration and watch position	Arm the watch-position event checking for an axis.	MAW Motion Arm Watch Position	Yes
	Disarm the watch-position event checking for an axis.	MDW Motion Disarm Watch Position	Yes
	Arm the servo-module registration-event checking for an axis.	MAR Motion Arm Registration	Yes
	Disarm the servo-module registration-event checking for an axis.	MDR Motion Disarm Registration	Yes
	Arm an output cam for an axis and output.	MAOC Motion Arm Output Cam	No
	Disarm one or all output cams connected to an axis.	MDOC Motion Disarm Output Cam	No
Tune an axis and run diagnostic tests for your control system. These tests include: · Motor/encoder hookup test · Encoder hookup test · Marker test	Use the results of an MAAT instruction to calculate and update the servo gains and dynamic limits of an axis.	MAAT Motion Apply Axis Tuning	No
	Run a tuning motion profile for an axis	MRAT Motion Run Axis Tuning	No
	Use the results of an MRHD instruction to set encoder and servo polarities.	MAHD Motion Apply Hookup Diagnostic	No
	Run one of the diagnostic tests on an axis.	MRHD Motion Run Hookup Diagnostic	No

If you want to	And	Use this instruction	Motion direct Command
Control multi-axis coordinated motion	Start a linear coordinated move for the axes of coordinate system.	MCLM Motion Coordinated Linear Move	No
	Start a circular move for the for the axes of coordinate system.	MCCM Motion Coordinated Circular Move	No
	Change in path dynamics for the active motion on a coordinate system.	MCCD Motion Coordinated Change Dynamics	No
	Stop the axes of a coordinate system.	MCS Motion Coordinated Stop	No
	Shutdown the axes of a coordinate system.	MCSD Motion Coordinated Shutdown	No
	Transition the axes of a coordinate system to the ready state and clear the axis faults.	MCSR Motion Coordinated Shutdown Reset	No
	Start a transform that links two coordinate systems together.	MCT() Motion Coordinated Transform	No
	Calculate the position of one coordinate system with respect to another coordinate system.	MCTP() Motion Calculate Transform Position	No

()You can use this instruction only with 1756-L6x controllers.

Motion Direct Command Dialog

You must be online to execute a Motion Direct Command. The online dialog has the Motion Group Shutdown and Execute buttons active. If you click either of these, action is taken immediately.



When the Motion Direct Command dialog is opened, focus is given to the Command Tree. In the Command list, you can either type the mnemonic and the list advances to the closest match or you can scroll down the list to select a command. Click the desired command and its dialog displays.

At the top of the dialog, in the title bar, there is a number at the end of the axis or group that the command is being applied upon. This is the Instance reference number. This number increases by one every time a command is accessed for that axis or group. The number is cleared when you execute RSLogix software.

Located at the bottom of the dialog are the following buttons: Motion Group Shutdown, Execute, Close, and Help.

Motion Group Shutdown Button

The Motion Group Shutdown button is located to the left of the screen to avoid accidental invoking of this command when you really want to execute the command accessed from the Command tree. Clicking on this button causes the Motion Group Shutdown instruction to execute. If you click on the Motion Group Shutdown button and it is successfully executed, a Result message is displayed in the results window below the dialog. Since the use of this button is an abrupt means of stopping motion, an additional message is displayed in the error text field. The message "MOTION GROUP SHUTDOWN executed!" is displayed with the intention of giving greater awareness of the execution of this command. If the command fails then an error is indicated as per normal operation. (See Error Conditions later in this chapter.)

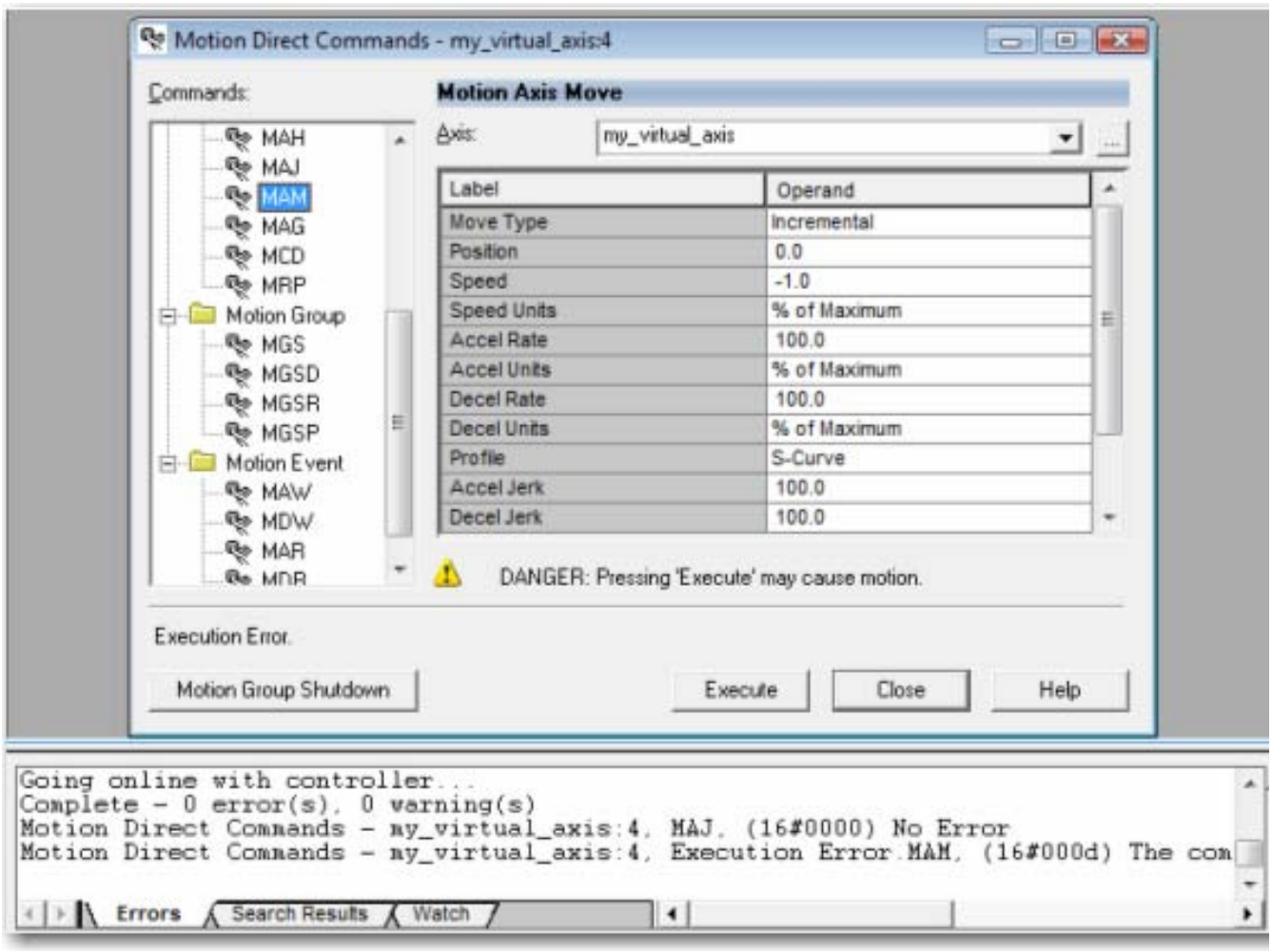
There is space above the Motion Group Shutdown button and below the line where status text is displayed when a command is executed.

Execute Button

Clicking the Execute button verifies the operands and initiates the current Motion Direct Command.

Motion Direct Command Error Process

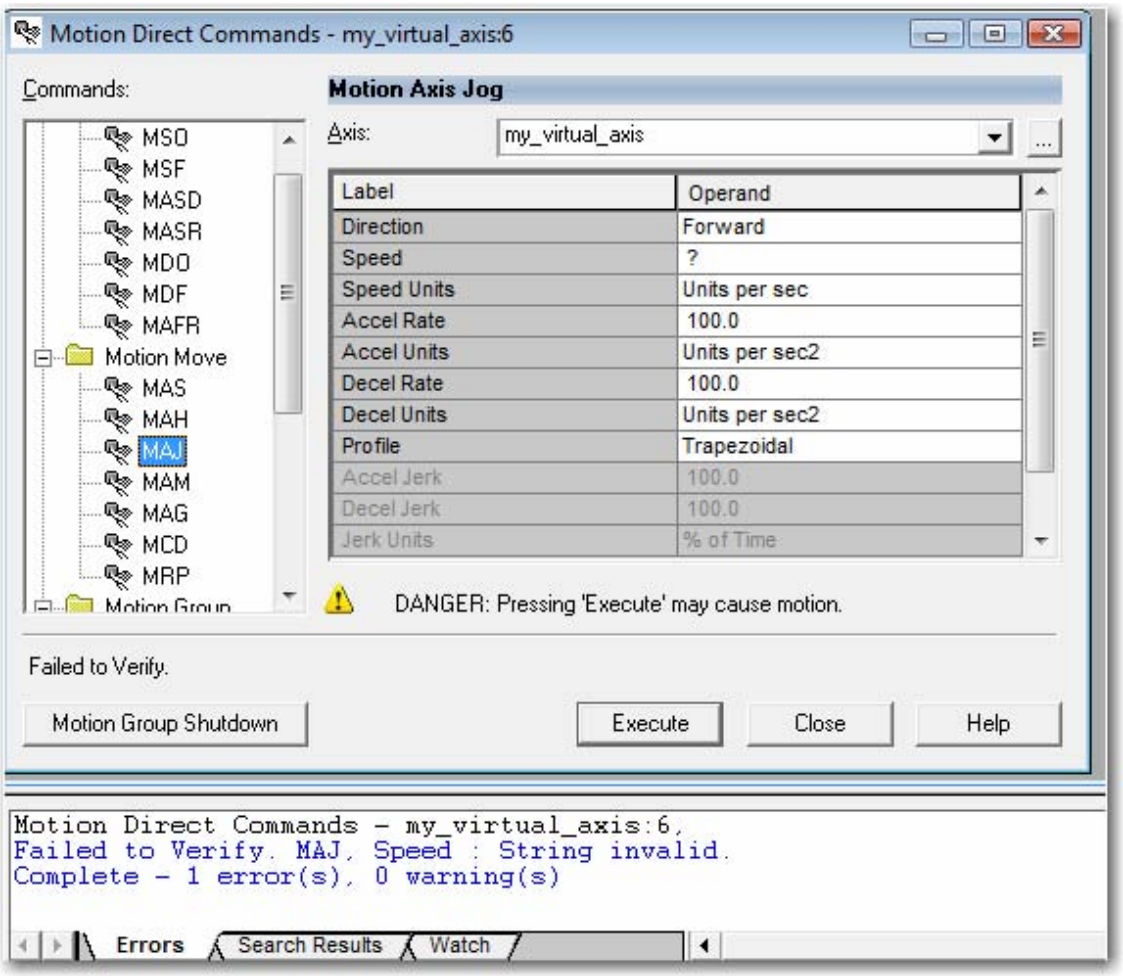
Whenever a Motion Direct Command is executed, there are two levels of error detection that are presented. The first level is verification of the command's operands. If a verification error is detected, a message "Failed to Verify" is posted on the dialog and an appropriate message is posted to the error result window. The second level is the initial motion direct command's error response return code. If an error code is detected, a message "Execution Error" is posted on the dialog.



Whether or not an error is detected, a detail message is displayed to the Error result window describing the results of the executed command.

Motion Direct Command Verification

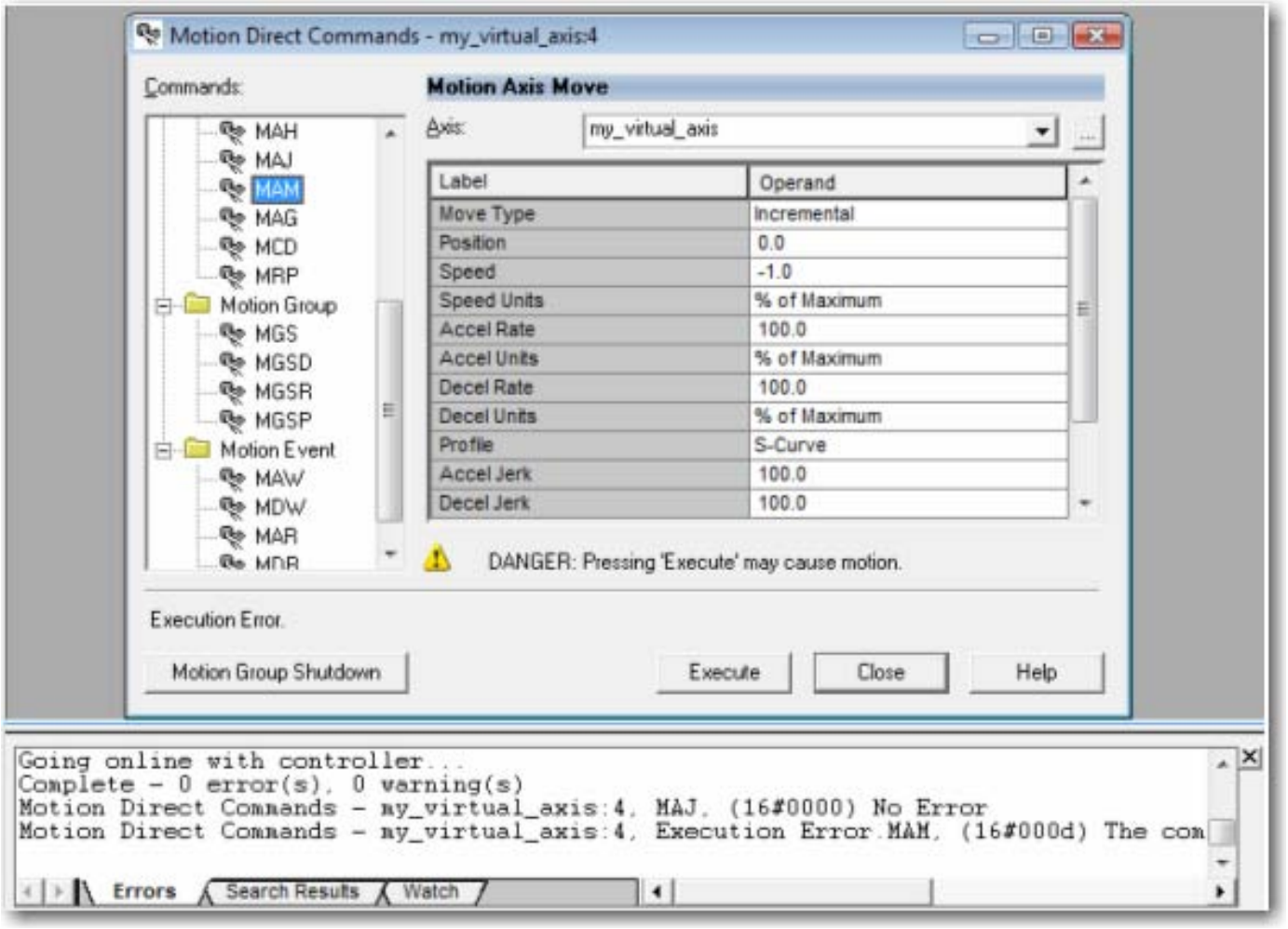
When you select Execute from a Motion Direct Command dialog, the operands are verified. If any operand fails verification, an error message "Failed to Verify" is displayed on the dialog and a detailed error message is displayed in the error result window describing the fault indicating the instance of Motion Direct Command that the results apply to. This allows multiple verification errors to be displayed and provides navigation to the error source, that is, double clicking the error in the results window will navigate to the appropriate Motion Direct Command dialog.



If no errors are detected during verification, then nothing is displayed.

Motion Direct Command Execution Error

When you select Execute from a Motion Direct Command dialog and the operands are verified as valid, then the command is executed. If the command fails immediately, then an error message "Execution Error" is displayed on the dialog. Whether or not an error is detected, a detailed message is displayed to the Error result window describing the immediate results of the executed command.



The message "Execution Error" is cleared on subsequent command execution or if a new command is selected from the command list.

The information pumped to the Error result window after an execution is not cleared. This allows for a history of what has been executed from a given instance of the Motion Direct Command dialog.

What If the Software Goes Offline or the Controller Changes Modes?

If RSLogix 5000 software transitions to offline, Hard Program mode (PROG), or Hard Run mode (RUN), then any executing Direct Command instruction continues execution and the Execute button is disabled.

Can Two Workstations Give Motion Direct Commands?

Whenever the Execute button is enabled and commands can be executed from a workstation, the group is locked. This means that another workstation cannot execute commands while this lock is in place. The lock stays in place until the workstation executing commands relinquishes the lock.

Handle Faults

Introduction

The controller has these types of motion faults.

Type	Description	Example
Instruction error	Caused by a motion instruction: · Instruction errors do not impact controller operation. · Look at the error code in the motion control tag to see why an instruction has an error. · Fix instruction errors to optimize execution time and make sure that your code is accurate	A Motion Axis Move (MAM) instruction with a parameter out of range
Fault	Caused by a problem with the servo loop: · You choose whether or not motion faults give the controller major faults. · Can shutdown the controller if you do not correct the fault condition	· Loss of feedback · Actual position exceeding an overtravel limit

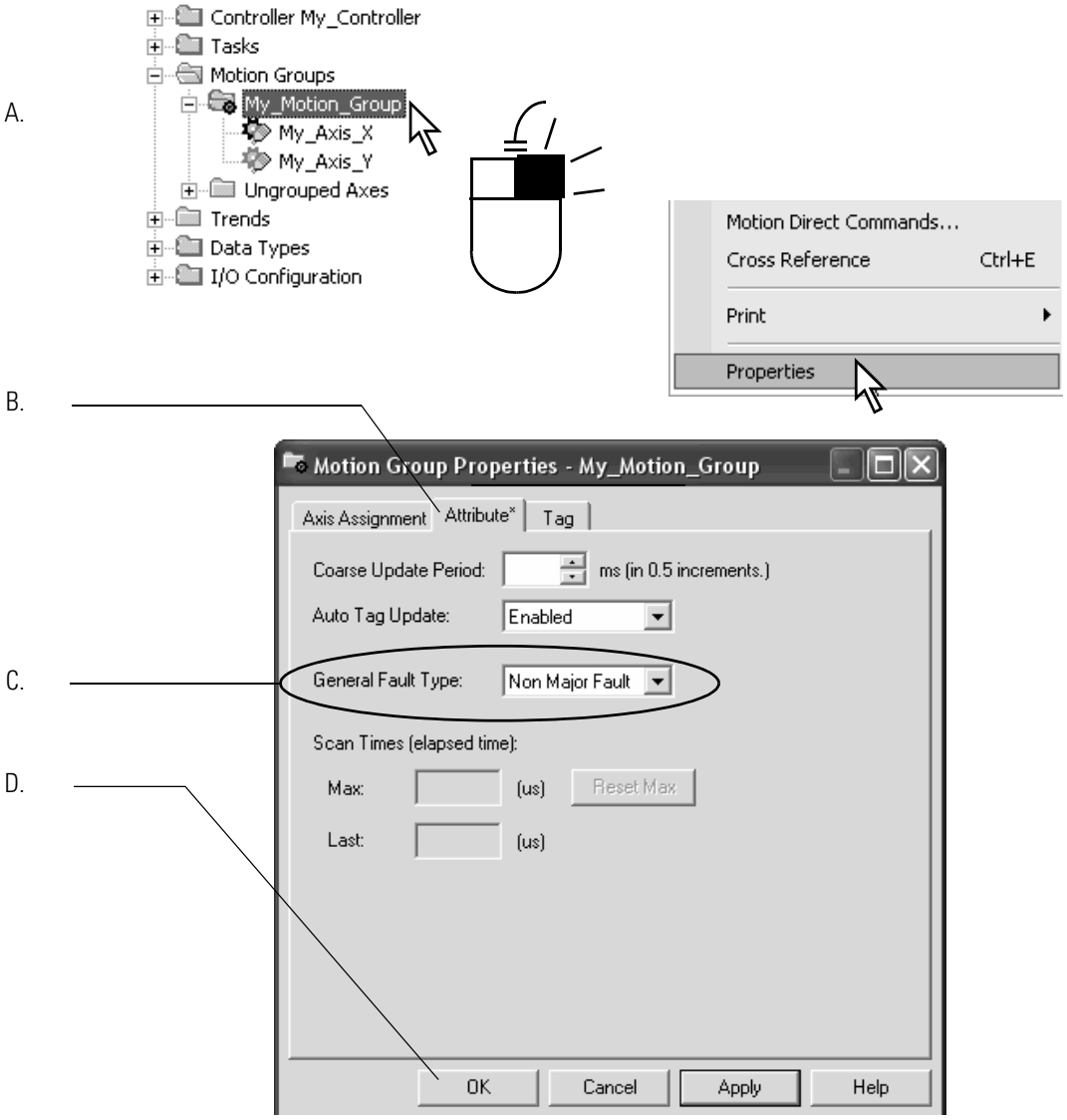
To handle motion faults:

- Choose If Motion Faults Shut Down the Controller
- Choose the Fault Actions for an Axis
- Set the Fault Action for an Axis

Choose If Motion Faults Shut Down the Controller

By default, the controller keeps running when there is a motion fault. As an option, you can have motion faults cause a major fault and shut down the controller.

Action	Details
1. Choose a General Fault Type.	Do you want any motion fault to cause a major fault and shut down the controller? · YES — Choose Major Fault. · NO — Choose Non-Major Fault. You must write code to handle motion faults.
2. Set the General Fault Type.	



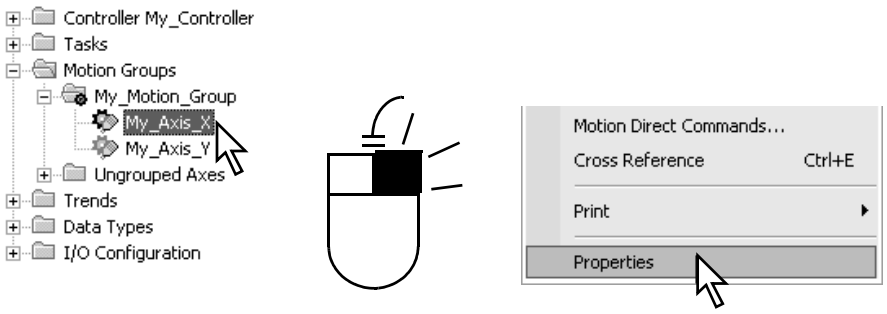
Choose the Fault Actions for an Axis

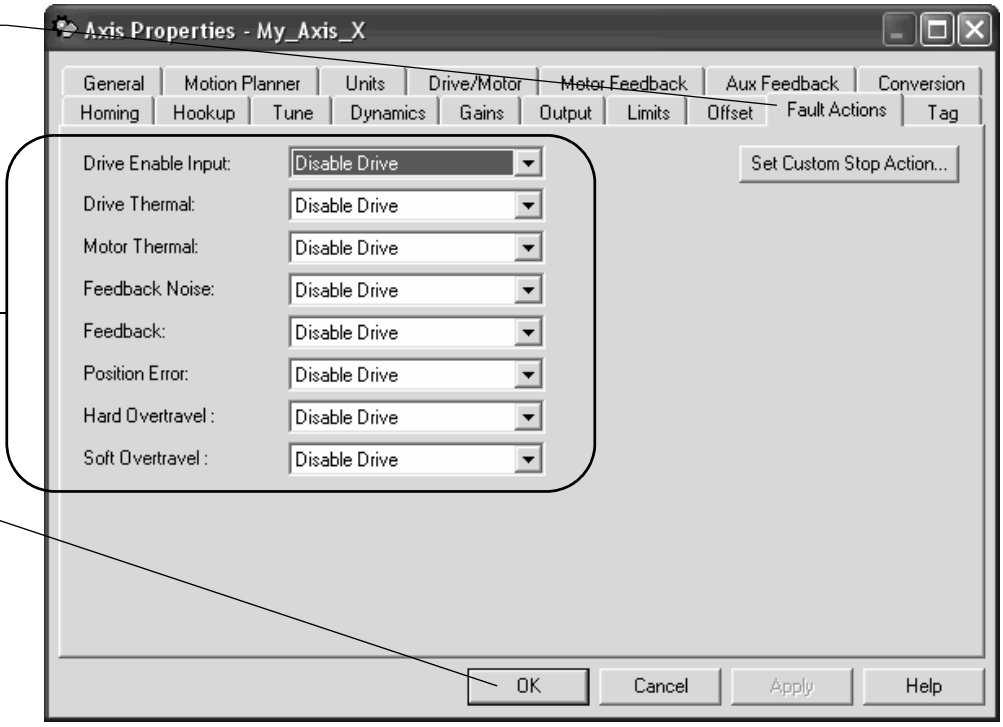
Use the fault actions to set how an axis responds to different types of faults. The type of faults depends on the type of axis and how you configure it.

If you want to	Then choose	Description	
Shutdown the axis and let it coast to a stop	Shutdown	Shutdown is the most severe action. Use it for faults that could endanger the machine or the operator if you don't remove power quickly and completely.	
		For this axis type	**When the fault happens**
		AXIS_SERVO	· Axis servo action is disabled. · The servo amplifier output is zeroed. · The appropriate drive enable output is deactivated. · The OK contact of the servo module opens. Use this to open the E-Stop string to the drive power supply.
		AXIS_SERVO_DRIVE	· Axis servo action and drive power structure are immediately disabled. · The axis coasts to a stop unless you use some form of external braking.
Disable the axis and let the drive stop the axis using it's best available stopping method	Disable Drive	**For this axis type**	**When the fault happens**
		AXIS_SERVO	· Axis servo action is disabled. · The servo amplifier output is zeroed. · The appropriate drive enable output is deactivated.
		AXIS_SERVO_DRIVE	· The drive switches to local servo loop control and the axis is slowed to a stop using the Stopping Torque. · If the axis doesn't stop in the Stopping Time, the servo action and the power structure are disabled.
Leave the servo loop on and stop the axis at its Maximum Deceleration rate	Stop Motion	Use this fault action for less severe faults. It is the gentlest way to stop. Once the axis stops, you must clear the fault before you can move the axis. The exception is Hardware Overtravel and Software Overtravel faults, where you can jog or move the axis off the limit.	
		For this axis type	**When the fault happens**
		AXIS_SERVO	The axis slows to a stop at the Maximum Deceleration Rate without disabling servo action or the servo module's Drive Enable output.
		AXIS_SERVO_DRIVE	· Control of the drive's servo loop is maintained. · The axis slows to a stop at the Maximum Deceleration rate without disabling the drive.
Write your own application code to handle the fault	Status Only	Use this fault action *only* when the standard fault actions are not appropriate. With this fault action, you must write code to handle the motion faults. For Stop Motion or Status Only, the drive must stay enabled for the controller to continue to control the axis. Selecting Status Only only lets motion continue if the drive itself is still enabled and tracking the command reference.	

Set the Fault Action for an Axis

Use the following steps to set the fault actions for an axis.

1. 

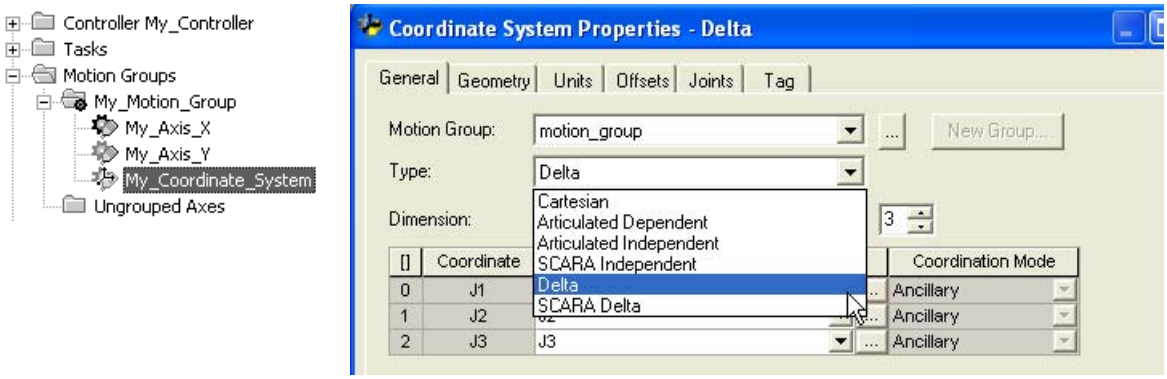
2. 3. 4. 

Create and Configure a Coordinate System

Introduction

In RSLogix 5000 software a coordinate system is a grouping of one or more primary and/or ancillary axes that you must create to generate coordinated motion. You can configure the coordinate system with one, two, or three dimensions. RSLogix 5000 software supports these types of geometry:

- Cartesian
- Articulated Dependant
- Articulated Independent
- Selective Compliant Assembly Robot Arm (SCARA) Independent
- Delta three-dimensional
- Delta two-dimensional
- SCARA Delta

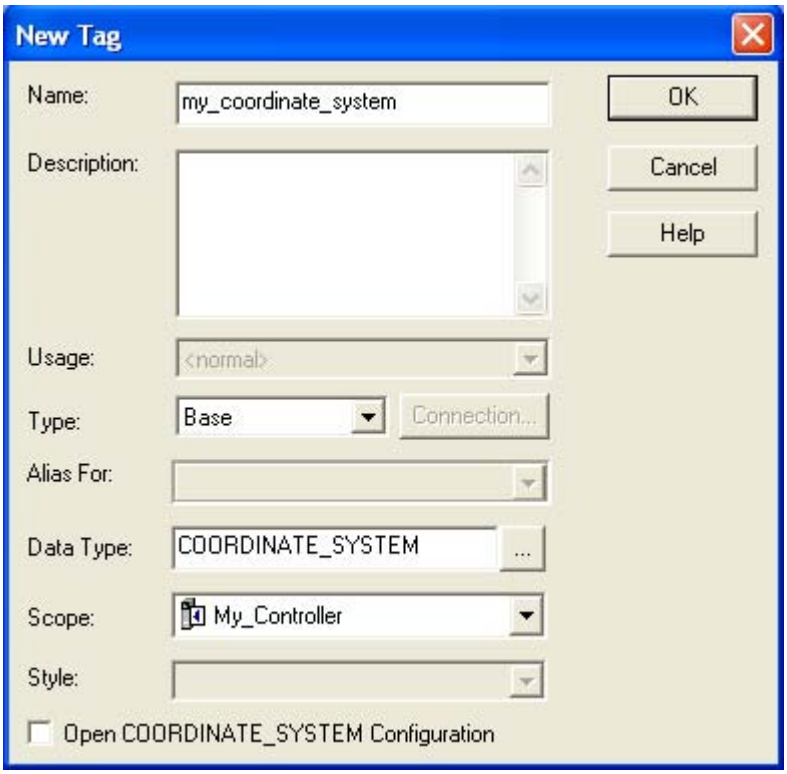


The Coordinate System tag is used to set the attribute values to be used by the Multi-Axis Coordinated Motion instructions in your motion applications. The Coordinate System tag must exist before you can run any of the Multi-Axis Coordinated Motion instructions. This is where you introduce the COORDINATE_SYSTEM data type, associate the coordinate system to a Motion Group, associate the axes to the coordinate system, set the dimension, and define the values later used by the operands of the Multi-Axis Motion Instructions. The values for Coordination Units, Maximum Speed, Maximum Acceleration, Maximum Deceleration, Actual Position Tolerance, and Command Position Tolerance are all defined by the information included when the Coordinate System tag is configured. This chapter describes how to name, configure, and edit your Coordinate System tag.

Create a Coordinate System

To create a coordinate system, right-click the motion group in the Controller Organizer and select New Coordinate System.

The New Tag dialog opens.



Enter Tag Information

A tag lets you allocate and reference data stored in the controller. A tag can be a single element, array, or a structure. With COORDINATE_SYSTEM selected as the Data Type, there are only two types of tags that you can create.

- A base tag lets you create your own internal data storage.
- An alias tag lets you assign a name of your choosing to an existing coordinate system tag.

New Tag Parameters

The following parameters appear on the New Tag dialog when you are creating a base tag or an alias tag.

Name

Enter a relevant name for the new tag. The name can be up to 40 characters and can be composed of letters, numbers, or underscores (_).

Description

Enter a description of the tag. This is an optional field and is used for annotating the tag.

Type

Use the dropdown menu to select what type of tag to create. For a coordinate system, the only valid choices are Tag and Alias. Selecting either Produced or Consumed generates an error when the OK button is pressed.

- Base refers to a normal tag (selected by default).
- Alias refers to a tag that references another tag with the same definition. Special parameters appear on the New Tag dialog that allow you to identify to which base tag the alias refers.

Alias For

If you selected Alias as the tag Type the Alias For field displays. Enter the name of the associated Base Tag.

Data Type

The Data Type field defines the size and layout of memory that is allocated when the tag is created. Select COORDINATE_SYSTEM.

Scope

Enter the Scope for the tag. The scope defines the range at which tags and routines can be created. A Coordinate System Tag can only be configured at the Controller Scope.

Style

The Style parameter is not activated. No entry for this field is possible.

After the information for the tag is entered, you have these options.

- Clicking OK, creates the tag and automatically places it in the Ungrouped Axes folder or the Motion Group if the tag was initiated from the Motion Group menu.
- Clicking the Configure button next to the Data Type field invokes the Coordinate System Tag Wizard to let you continue to configure the Coordinate System tag.

Open COORDINATE_SYSTEM Configuration

When checked, displays the wizard screens that guide you through the process of configuring a coordinate system.

Coordinate System Wizard Dialogs

The Coordinate System Wizard dialogs are the same dialogs that appear when you access Coordinate System Properties but instead of appearing as tabbed dialogs, they advance you through the process by individual dialogs. At the bottom of each dialog, is a series of buttons. To advance to the next dialog, click Next and the information you entered is saved and you advance to the next Wizard dialog. To end your progression through the Wizard dialogs click Finish. The information entered to this point is saved and the coordinate system is stored in the Controller Organizer under either the Ungrouped Axes folder or the Motion Group (if a motion group has been associated with the coordinate system).

It is not necessary to use the Wizard dialogs to configure your coordinate system. Once it has been created, you can access the Coordinate System Properties dialog and enter the information for the coordinate system. See the section entitled Editing Coordinate System Properties later in this manual for detailed information about entering configuration information.

General Wizard Dialog

The General dialog lets you:

- associate the tag to a Motion Group.
- enter the coordinate system type.
- select the Dimension for the tag (that is, the number of associated axes).
- specify the number of dimensions to transform.
- enter the associated axis information.
- choose whether to update Actual Position values of the coordinate system automatically during operation.

This dialog has the same fields as the General tab found under Coordinate System Properties.

Geometry Wizard Dialog

The Geometry dialog lets you configure key attributes related to non-Cartesian geometry and shows the bitmap of the associated geometry.

Offsets Wizard Dialog

The Offset dialog lets you configure the offsets for the base and end effector. This dialog shows the bitmaps for the offsets related to the geometry.

Units Wizard Dialog

The Units dialog lets you determine the units that define the coordinate system. At this dialog you define the Coordination Units and the Conversion Ratios. This dialog has the same fields as the Units tab found under Coordinate System Properties.

Dynamics Wizard Dialog

Use the Dynamics dialog for entering the Vector values used for Maximum Speed, Maximum Acceleration, and Maximum Deceleration. It is also used for entering the Actual and Command Position Tolerance values. This dialog has the same fields as the Dynamics tab found under Coordinate System Properties.

Manual Adjust Button

The Manual Adjust button is inactive when creating a Coordinate System tag via the Wizard dialogs. It is active on the Dynamics tab of the Coordinate System Properties dialog. It is described in detail in the Editing Coordinate System Properties later in this chapter.

Tag Wizard Dialog

The Tag dialog lets you rename your Tag, edit your description, and review the Tag Type, Data Type, and Scope information.

The only fields that you can edit on the Tag dialog are Name and Description. These are the same fields as on the New Tag dialog and the Coordinate System Properties Tag tab.

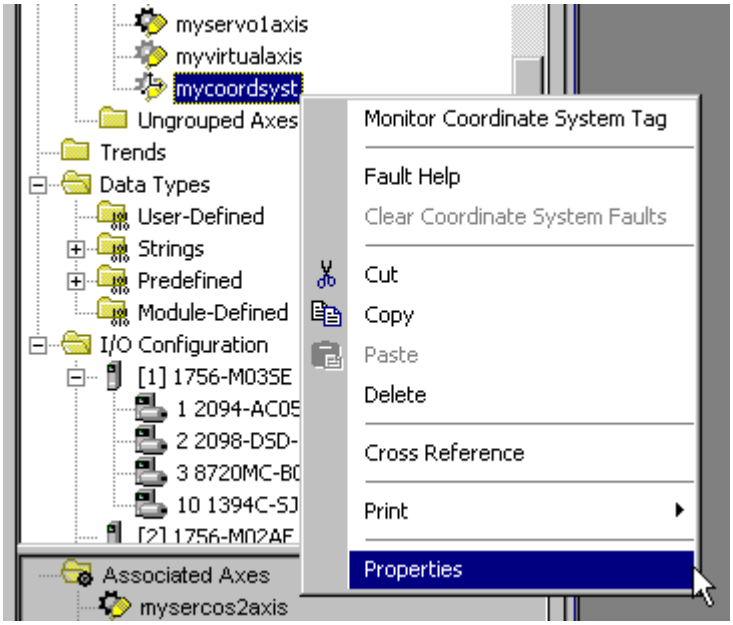
Edit Coordinate System Properties

Create your Coordinate System in the New Tag dialog, then configure it. If you did not use the Wizard dialogs available from the Configure button on the New Tag dialog, you can make your configuration selections from the Coordinate System Properties dialog. You can also use the Coordinate System Properties dialogs to edit an existing Coordinate System tag. These have a

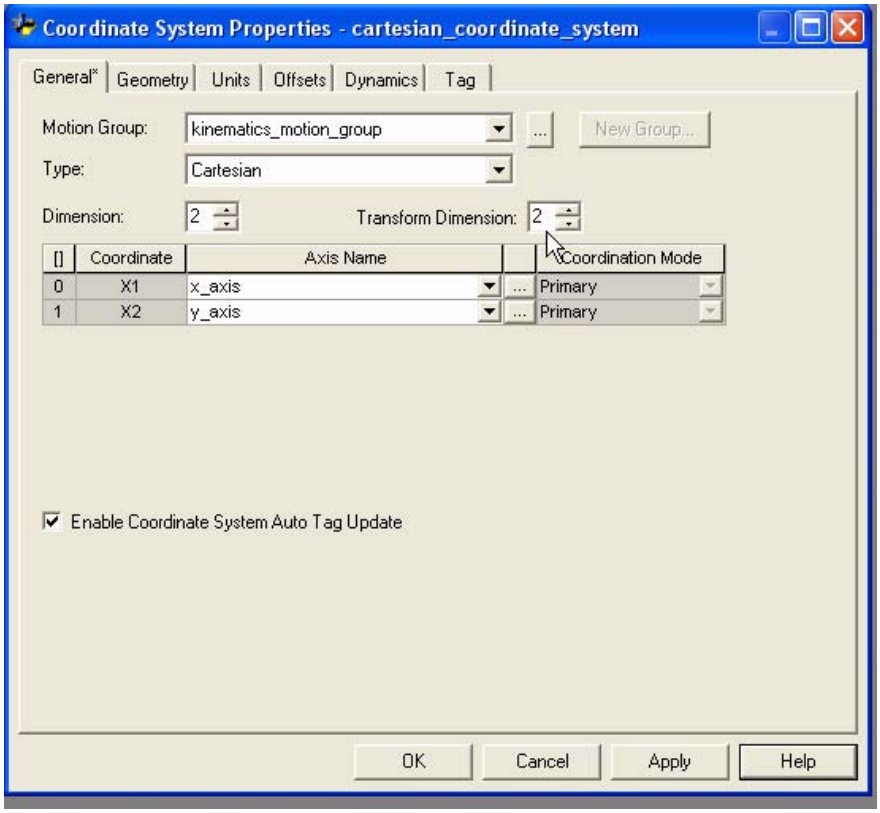
series of tabs that access a specific dialog for configuring the different facets of the Coordinate System. Make the appropriate entries for each of the fields. An asterisk appears on the tab to indicate changes have been made but not implemented. Click Apply to save your selections.

TIP When you configure your coordinate system, some fields may be unavailable (dimmed) because of choices you made in the New Tag dialog.

In the Controller Organizer, right-click the coordinate system to edit and select Coordinate System Properties from the pull-down menu.



The Coordinate System Properties General dialog appears. The name of the Coordinate System tag that is being edited appears in the title bar to the right of Coordinate System Properties. The General tab dialog for a Cartesian coordinate system is shown below.



General Tab

Use this tab to do the following for a coordinate system:

- Assign the coordinate system, or terminate the assignment of a coordinate system, to a Motion Group.
- Choose the type of coordinate system you are configuring.
- Change the number of dimensions, that is, the number of axes.
- Specify the number of axes to transform.
- Assign axes to the coordinate system tag.
- Enable/Disable automatic updating of the tag.

  RSLogix 5000 software supports only one Motion Group tag per controller.

Motion Group

The Motion Group button selects and displays the Motion Group to which the Coordinate System is associated. A Coordinate System assigned to a Motion Group appears in the Motion Groups branch of the Controller Organizer, under the selected Motion Group sub-branch. Selecting <none> terminates the Motion Group association, and moves the coordinate system to the Ungrouped Axes sub-branch of the Motions Groups branch.

Ellipsis (…) Button

The Ellipsis button opens the Motion Group Properties dialog for the Assigned Motion Group, where you can edit the Motion Group properties. If no Motion Group is assigned to this coordinate system, this button is unavailable.

New Group Button

The New Group button opens the New Tag dialog, where you can create a new Motion Group tag. This button is enabled only if no Motion Group tag has been created.

Type

Type selects and displays the type of coordinate system (robot type) in the Motion Group. Available choices are Cartesian, Articulated Dependent, Articulated Independent, SCARA Independent, Delta, and SCARA Delta. The type of coordinate system you choose in this field changes the configuration tabs that are available.

Dimension

Enter the coordinate system dimensions, that is, the number of axes, that this coordinated system is to support. The options are 1, 2, or 3 in keeping with its support of a maximum of three axes. Changes in the Dimension spin also reflect in the Axis Grid by either expanding or contracting the number of fields available. Data is set back to the defaults for any axis that is removed from the Axis Grid due to reducing the Dimension field.

Transform Dimension

Enter the number of axes in the coordinate system that you want to transform. The options are 1, 2, or 3 in keeping with its support of a maximum of 3 axes. The number of axes that you transform must be equal to or less than the specified coordinate system dimensions. The transform function always begins at the first axis. For example, if you have specified that the coordinate system has 3 axes but indicate only that 2 axes be transformed, then axes 1 and 2 will

be transformed. In other words, you cannot specify that only axes number 2 and number 3 be transformed.

Axis Grid

The Axis Grid is where you associate axes to the Coordinate System. There are five columns in the Axis Grid that provide information about the axes in relation to the Coordinate System.

[] (Brackets)

The Brackets column displays the indices in tag arrays used with the current coordinate system. The tag arrays used in multi-axis coordinated motion instructions map to axes using these indices.

Coordinate

The text in this column X1, X2, or X3 (depending on the entry to the Dimension field) is used as a cross reference to the axes in the grid. For a Cartesian system the mapping is simple.

Axis Name

The Axis Name column is a list of combo boxes (the number is determined by the Dimension field) used to assign axes to the coordinate system. The pull-down lists display all of the Base Tag axes defined in the project. (Alias Tag axes do not display in the pull-down list.) They can be axes associated with the motion group, axes associated with other coordinated systems, or axes from the Ungrouped Axes folder. Select an axis from the pull-down list. The default is <none>. It is possible to assign fewer axes to the coordinate system than the Dimension field allows; however, you will receive a warning when you verify the coordinate system and, if left in that state, the instruction generates a run-time error. You can assign an axis only once in a coordinate system. Ungrouped axes also generate a runtime error.

Ellipsis Button (...)

The Ellipsis buttons in this column take you to the Axis Properties pages for the axis listed in the row. See the Creating and Configuring Your Motion Axis chapter in this manual for information about the Axis Properties page.

Coordination Mode

The Coordination Mode column indicates the axes that are used in the velocity vector calculations. If the type of coordinate system is specified as Cartesian, then Primary axes are used in these calculations. For

non-Cartesian coordinate systems, the coordination mode for the axes defaults to Ancillary.

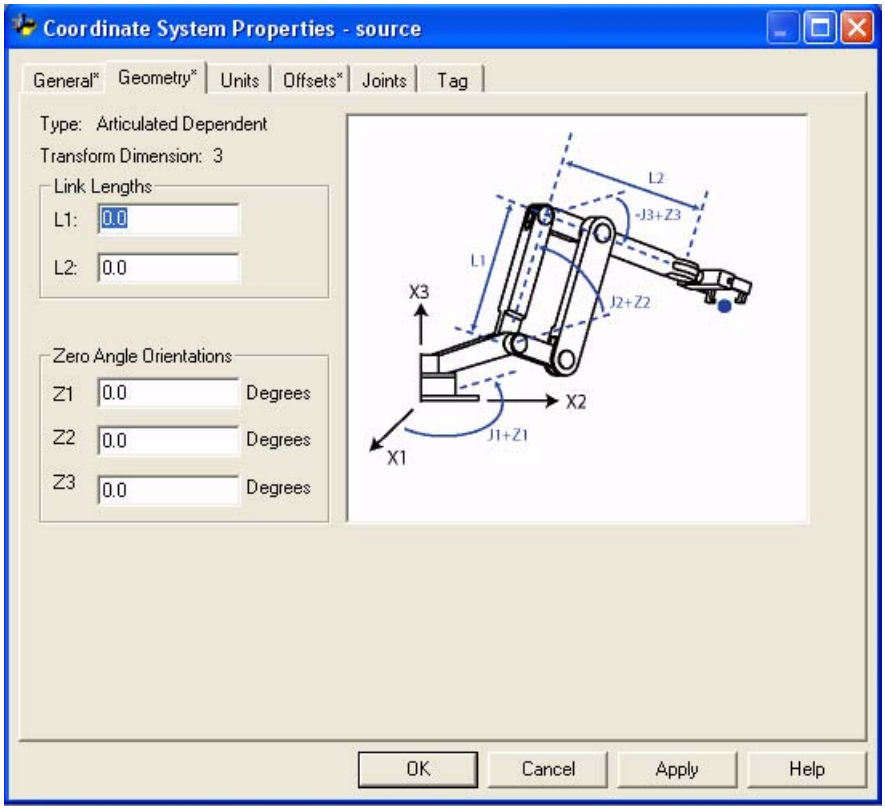
Enable Coordinate System Auto Tag Update

The Enable Coordinate System Auto Tag Update checkbox lets you determine whether the Actual Position values of the current coordinated system are automatically updated during operation. Click the checkbox to enable this feature. The Coordinate System Auto Tag Update feature can ease your programming burden if you would need to add GSV statements to the program in order to get the desired result. However, by enabling this feature the Coarse Update rate is increased. Whether to use the Coordinate System Auto Tag Update feature depends upon the trade-offs between ease in programming and increase in execution time. Some users may want to enable this feature in the initial programming of their system to work out the kinks and then disable it and enter the GSV statements to their program to lower their execution time.

Enabling this feature may result in some performance penalty.

Click Apply to implement your entries or cancel to not save the new entries.

To edit the Geometry parameters for the robotic arm, select the Geometry tab.



Geometry Tab

The Geometry tab of the Coordinate System Properties is where you can specify the link lengths and zero angle orientation values for articulated robotic arms.

The graphic displayed on this tab shows a typical representation of the type of coordinate system you selected on the General tab. Your robot should look similar to the one shown in the graphic but may be somewhat different depending on your application.

Link Lengths Box

The Link Length box displays fields to let you specify a value for the length of each link in an articulated robotic arm (coordinate system). The measurement units for the articulated coordinate system are defined by the measurement units configured for the affiliated Cartesian coordinate system. The two coordinate systems are linked or affiliated with each other by an MCT instruction.

When specifying the link length values be sure that the values are calculated using the same measurement units as the linked Cartesian coordinate system. For example, if the manufacturer specifies the robot link lengths using millimeter units and you want to configure the robot using inches, then you must convert the millimeter link measurements to inches and enter the values in the appropriate link length fields.

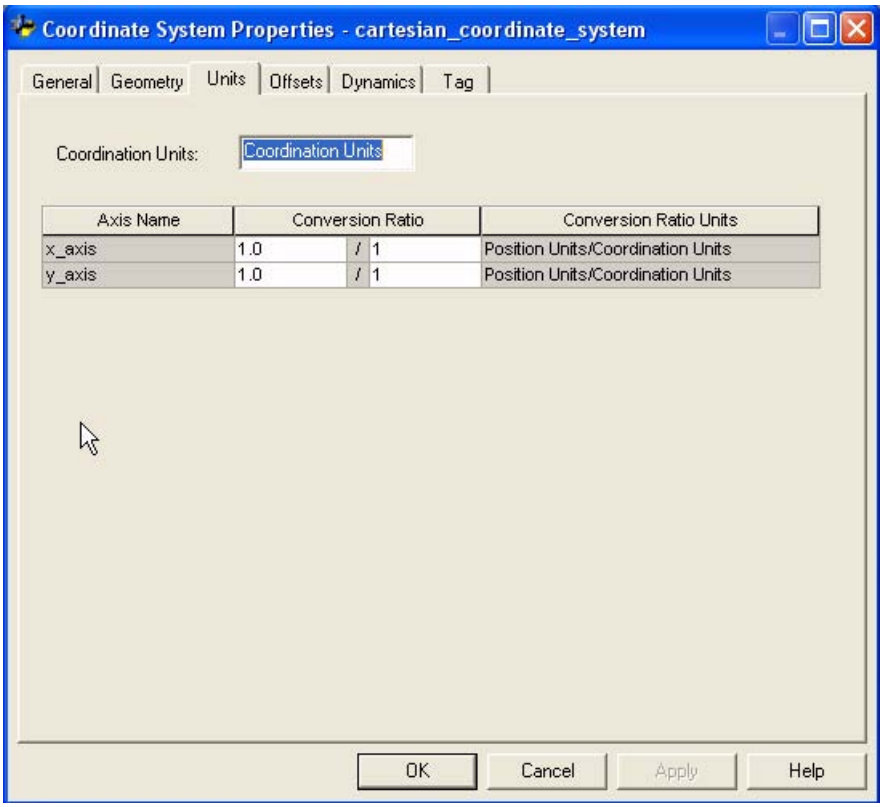
IMPORTANT Be sure that the link lengths specified for an articulated coordinate system are in the same measurement units as the affiliated Cartesian coordinate system. Your system will not work properly if you are using different measurement units.

The number of fields available for configuration in the link lengths box is determined by values entered on the General tab for the type of coordinate system, total coordinate system dimensions, and transform dimensions. The link identifiers are L1 and L2 in the corresponding graphic. These fields are not configurable for a Cartesian coordinate system.

Zero Angle Orientations Box

The zero-angle orientation is the rotational offset of the individual joint axes. If applicable, enter the offset value in degrees for each joint axis. The number of available fields is determined by the coordinate dimension value entered on the General tab. The angle identifiers are Z1, Z2, and Z3 in the corresponding graphic.

To edit the Units properties, select the Units tab to access the Coordinate System Properties Units dialog.



Units Tab

The Units tab of the Coordinate System Properties is where you determine the units that define the coordinate system. This dialog is where you define the Coordination Units and the Conversion Ratios.

Coordination Units

The Coordination Units field lets you define the units to be used for measuring and calculating motion related values such as position and velocity. The coordination units do not need to be the same for each coordinate system. Enter units that are relevant to your application and maximize ease of use. When you change the Coordination Units, the second portion of the Coordination Ratio Units automatically changes to reflect the new units. Coordination Units is the default.

Axis Grid

The Axis Grid of the Units dialog displays the axis names associated with the coordinate system, the conversion ratio, and the units used to measure the conversion ratio.

Axis Name

The Axis Name column contains the names of the axes assigned to the coordinate system in the General dialog. These names appear in the order that they were configured into the current coordinate system. This column is not editable from this dialog.

Conversion Ratio

The Conversion Ratio column defines the relationship of axis position units to coordination units for each axis. For example, if the position units for an axis is in millimeters and the axis is associated with a coordinate system whose units are in inches, then the conversion ratio for this axis/coordinate system association is 25.4/1 and can be specified in the appropriate row of the Axis Grid.

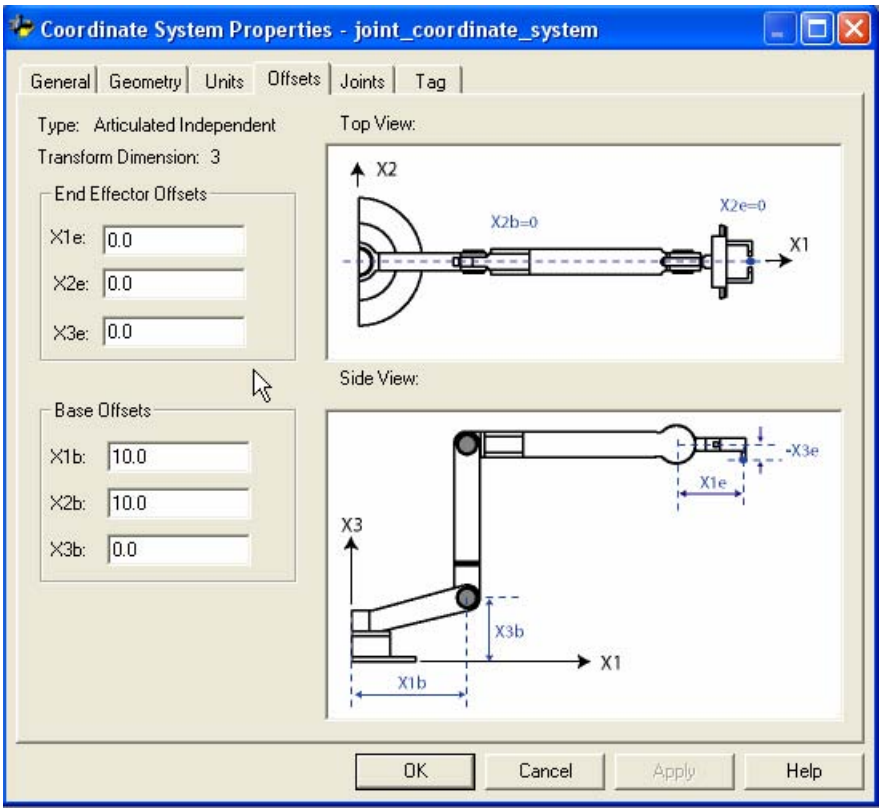
The numerator can be entered as a float or an integer. The denominator must be entered as an integer only.

Conversion Ratio Units

The Conversion Ratio Units column displays the axis position units to coordination units used. The Axis Position units are defined in the Axis Properties – Units dialog and the coordination units are defined in Coordinated System Properties – Units dialog. These values are dynamically updated when changes are made to either axis position units or coordination units.

Click Apply to preserve your edits or Cancel to discard your changes.

Click the Offsets tab to access the Coordinate System Properties Offset dialog.



Offsets Tab

The Offsets tab of the Coordinate System Properties dialog is where you define the end effector and base offset values for the robotic arm. This tab shows the top and/or sides view of a typical robotic arm based on the type of coordinate system and coordinate Transform dimension values specified on the General tab. The number of available offset fields in each box is determined by the number of axes associated with the coordinate system.

When specifying the end effector and base offset values be sure that the values are calculated using the same measurement units as the linked Cartesian coordinate system. For example, if the manufacturer specifies the robot offset using millimeter units and you want to configure the robot using inches then, you must convert the millimeter link measurements to inches and enter the values in the appropriate offset fields.

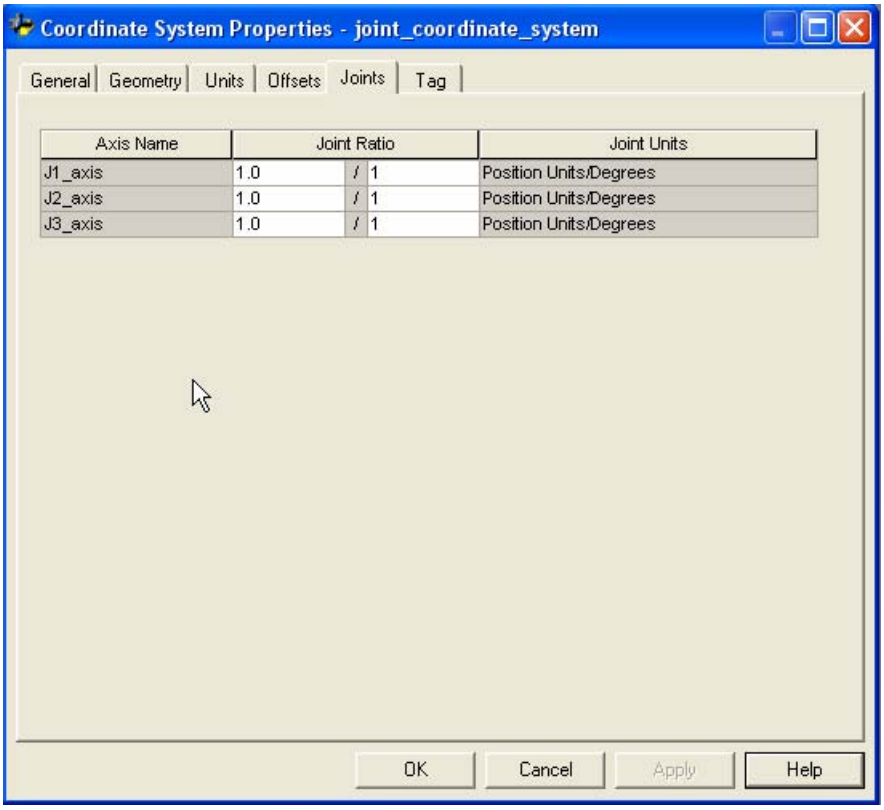
End Effector Offsets Box

The end effector offset value specifies the dimensions of the end effector. The correct end effector offsets are typically available from the manufacturer. The end effector indicators are X1e, X2e, and X3e in the corresponding graphic.

Base Offsets Box

The RSLogix 5000 Kinematics internal equations define the robot origin relative to the first joint of the robotic arm. The robot manufacturer may specify the origin at a different location. The difference between these two locations is the base offsets value. The correct base offset values are typically available from the robot manufacturer. The base offset indicators are X1b, X2b and X3b in the corresponding graphic.

If you are configuring an articulated coordinate system, click the Joints tab to access the Coordinate System Properties Joints dialog.



Joints Tab

The Joints tab is accessible only if you are configuring or editing an articulated coordinate system. This dialog is where you define the Joint Conversion Ratios. Joint axis units are always specified in degrees.

Axis Name

The Axis Name column displays the names of the axes associated to the coordinate system. The names appear in the order that they were configured into the coordinate system. This is a read-only field.

Joint Ratio

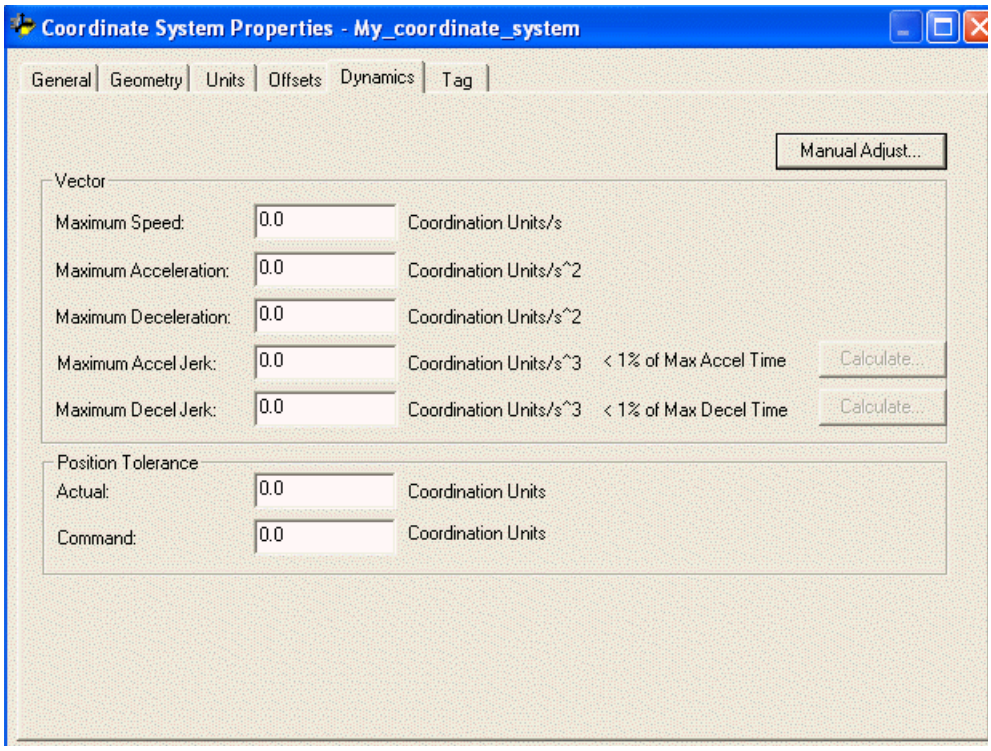
The Joint Ratio column (shown in white) is divided into two columns that define the relationship between the axis position units to the joint axis units.

The left-half of the Joint Ratio column is a configurable field that lets you specify a value for the axis position units (numerator). The right-half of the Joint Ratio column is a configurable field that lets you specify a value for the joint axis units (denominator). Keep in mind that Joint axis units are always specified as degrees.

Joint Units

The Joint Units column is a read-only field that displays the configured axis position units to the joint units. The Axis Position units are defined in the Axis Properties – Units dialog. Joint units are always defined as degrees.

If you are configuring a Cartesian coordinate system, click the Dynamics tab to access the Coordinate System Properties Dynamics dialog.



Dynamics Tab

The Dynamics dialog is accessible only if you are configuring a Cartesian coordinate system. The Dynamics tab is for entering the Vector values used for Maximum Speed, Maximum Acceleration, Maximum Deceleration, Maximum Acceleration Jerk and Maximum Deceleration Jerk. It is also used for entering the Actual and Command Position Tolerance values.

Vector Box

In the Vector box, values are entered for Maximum Speed, Maximum Acceleration, Maximum Deceleration, Maximum Acceleration Jerk, and

Maximum Deceleration Jerk. The values are used by the Coordinated Motion instructions in calculations when their operands are expressed as percent of Maximum. The Coordination Units to the right of the edit boxes automatically change when the coordination units are redefined at the Units dialog.

Maximum Speed

Enter the value for Maximum Speed to be used by the Coordinated Motion instructions in calculating vector speed when speed is expressed as a percent of maximum.

Maximum Acceleration

Enter the value for Maximum Acceleration to be used by the Coordinated Motion instructions to determine the acceleration rate to apply to the coordinate system vector when acceleration is expressed as a percent of maximum.

Maximum Deceleration

Enter the value for Maximum Deceleration to be used by the Coordinated Motion instructions to determine the deceleration rate to apply to the coordinate system vector when deceleration is expressed as a percent of maximum. The Maximum Deceleration value must be a nonzero value to achieve any motion using the coordinate system.

Maximum Acceleration Jerk

The jerk parameters only apply to S-curve profile moves using these instructions:

- MCS
- MCCD
- MCCM
- MCLM

The Maximum Acceleration Jerk rate of the coordinate system, in Coordination Units/second^3, defaults to 100% of the maximum acceleration time. The speed and acceleration rate for this calculation are defined above.

$$\frac{\text{MaxAccel}^2}{\text{Speed}} = \text{Maximum Acceleration Jerk}$$

The Maximum Accel Jerk value entered is used when the motion instruction is set with Jerk Units=% of Maximum. When a Multi-axis Motion Instruction has Jerk Units=units per sec^3 then the maximum

acceleration jerk value is derived from the motion instruction faceplate. The jerk units for the motion instruction also allow for Jerk Units=% of Time, with 100% of Time. This means that the entire S-curve move will have Jerk limiting. This is the default mode. An S-curve move with 0% of Time will result in a trapezoidal profile, and have 0% Jerk limiting. If set manually, enter the value in units=Coordination Units/second^3 units. You can also use the Calculate button to view this value in terms of units=% of Time.

Maximum Deceleration Jerk

The jerk parameters only apply to S-curve profile moves using these instructions:

- MCS
- MCCD
- MCCM
- MCLM

The Maximum Deceleration Jerk rate of the coordinate system, in Coordination Units/second^3, defaults to 100% of the maximum deceleration time. The speed and deceleration rate for the calculation are defined above.

$$\frac{\text{MaxDecel}^2}{\text{Speed}} = \text{Maximum Deceleration Jerk}$$

The Maximum Decel Jerk value entered is used when the motion instruction is set with Jerk Units=% of Maximum. When a Multi-axis motion instruction has Jerk Units=units per sec^3 then the Max Deceleration Jerk value is derived from the Motion Instruction faceplate. The jerk units for the motion instruction also allow for Jerk Units=% of Time, with 100% of Time meaning the entire S-curve move will have Jerk limiting, which is the default mode. An S-curve move with 0% of Time will result in a trapezoidal profile, and have 0% Jerk limiting. If set manually, enter the value in units=Coordination Units/second^3 units. You can also use the optional Calculate button to view the value in terms of units=% of Time.

Position Tolerance Box

In the Position Tolerance Box, values are entered for Actual and Command Position Tolerance values. See the Logix5000 Motion Instruction Set Reference Manual, publication number 1756-RM007, for more information regarding the use of Actual and Command Position Tolerance.

Actual

Enter the value in coordination units, for Actual Position to be used by Coordinated Motion instructions when they have a Termination Type of Actual Tolerance.

Command

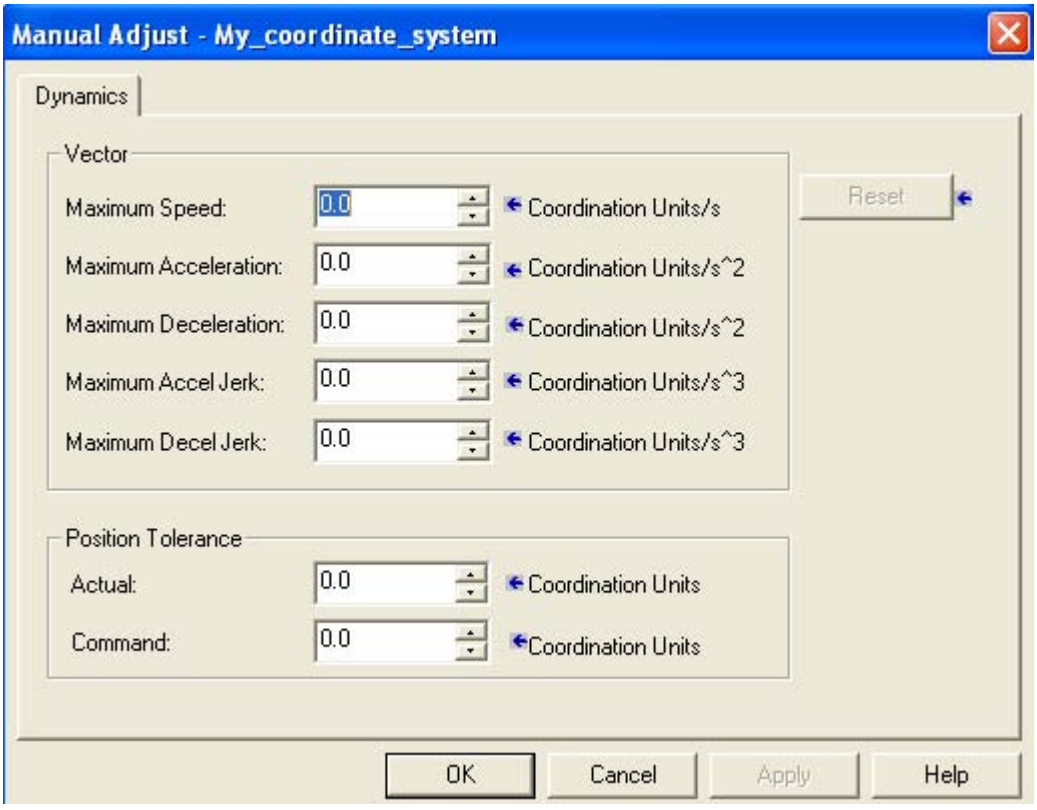
Enter the value in coordination units, for Command Position to be used by Coordinated Motion instructions when they have a Termination Type of Command Tolerance.

Manual Adjust Button

The Manual Adjust button on the Coordinate System Dynamics tab accesses the Manual Adjust Properties dialog. The Manual Adjust button is enabled only when there are no pending edits on the properties dialog.

Dynamics Tab Manual Adjust

At this dialog you can make changes to the Vector and Position Tolerance values.



These changes can be made either online or offline. The blue arrows to the right of the fields indicate that they are immediate commit fields. This means

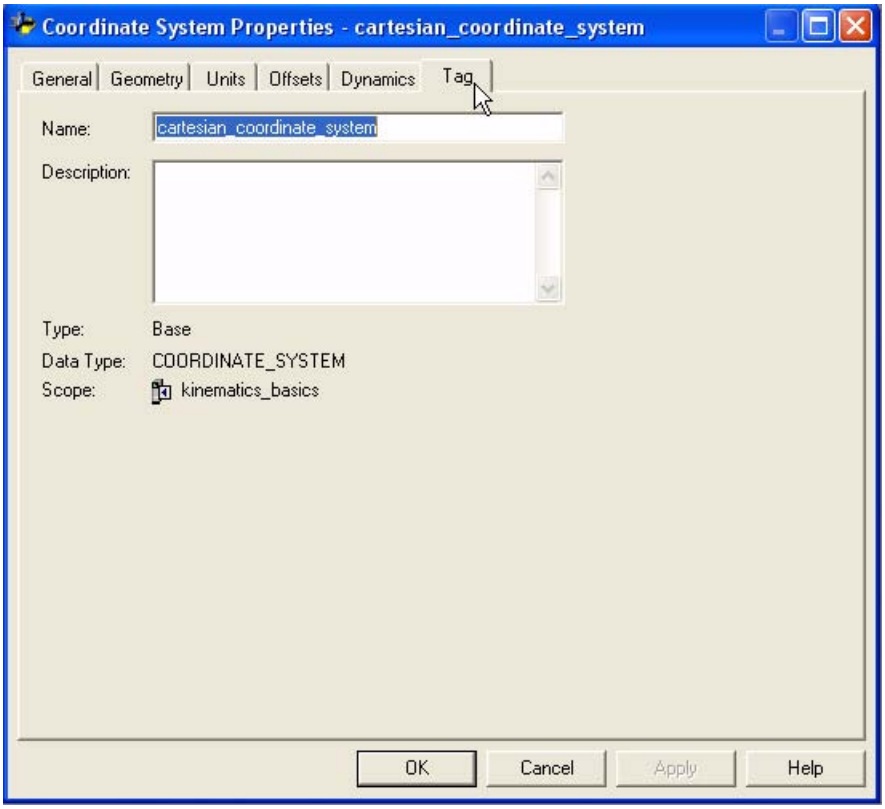
that the values in those fields are immediately updated to the controller if online or to the project file if offline.

Reset Button

The Reset Button reloads the values that were present at the time this dialog was entered. The blue arrow to the right of the Reset button means that the values are immediately reset when the Reset button is clicked.

Tag Tab

The Tag tab is for reviewing your Tag information and renaming the tag or editing the description.



Use this tab to modify the name and description of the coordinate system. When you are online, all of the parameters on this tab transition to a read-only state, and cannot be modified. If you go online before you save your changes, all pending changes revert to their previously-saved state.

Name

Name displays the name of the current tag. You can rename the tag at this time. The name can be up to 40 characters and can include letters, numbers, and underscores (_). When you rename a tag, the new name replaces the old one in the Controller Organizer after click OK or Apply.

Description

Description displays the description of the current tag, if any is available. You can edit this description. The edited description replaces the existing description when you click OK or Apply.

Tag Type

Tag Type indicates the type of the current Coordinate System tag. This type may be:

- Base
- Alias

The field is not editable and is for informational purposes only.

Data Type

Data Type displays the data type of the current Coordinate System tag, which is always COORDINATE_SYSTEM. This field cannot be edited and is for informational purposes only.

Scope

Scope displays the scope of the current Coordinate System tag. The scope for a Coordinate System tag can be only controller scope. This field is not editable and is for informational purposes only.

Notes:

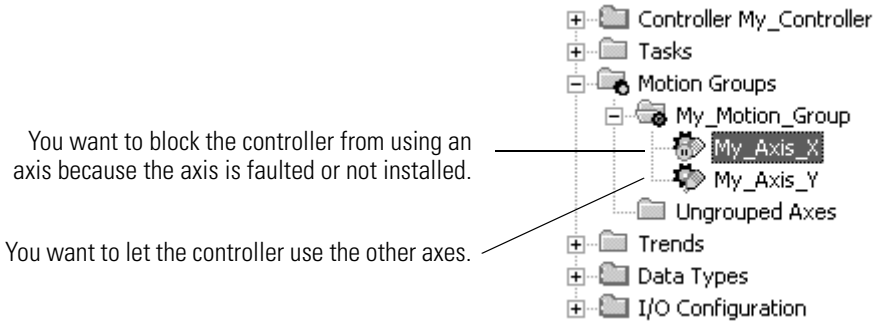
Inhibit an Axis

Introduction

Use this chapter to block the controller from using an axis.

When to Inhibit an Axis

Use the following information to determine when to inhibit an axis.



Example 1

Suppose you make equipment that has between 8 and 12 axes, depending on which options your customer buys. In that case, set up one project for all 12 axes. When you install the equipment for a customer, inhibit those axes that the customer didn't buy.

Example 2

Suppose you have two production lines that use the same SERCOS ring. And suppose one of the lines gets a fault. In that case, inhibit the axes on that line. This lets you run the other line while you take care of the fault.

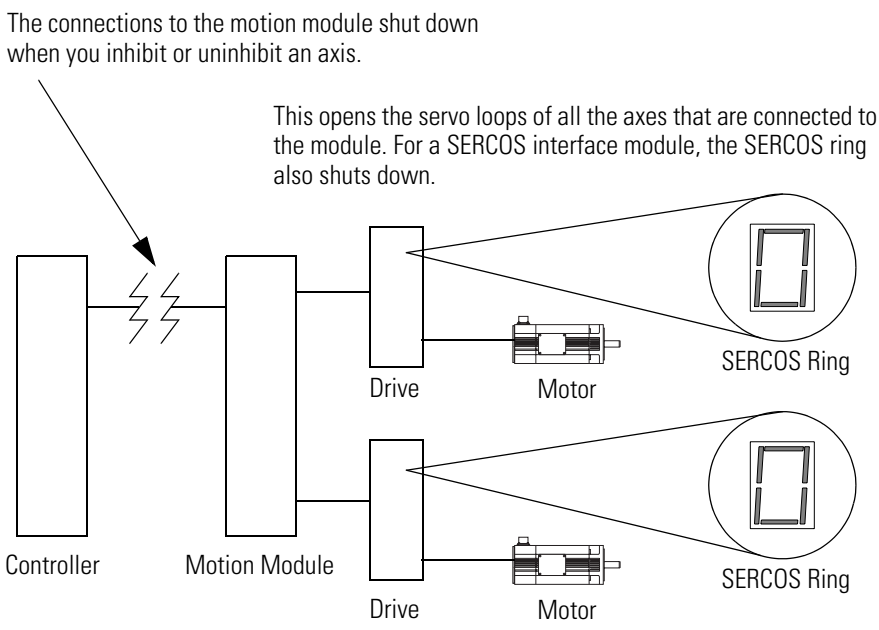
Before You Begin

Before you inhibit or uninhibit an axis, turn off all of the axes.

Before you inhibit or uninhibit an axis:

1. Stop all motion.
2. Open the servo loops of all the axes. Use an instruction such as the Motion Servo Off (MSF) instruction.

This lets you stop motion under your control. Otherwise the axes turn off on their own when you inhibit or uninhibit one of them.


The connections to the motion module shut down when you inhibit or uninhibit an axis.
This opens the servo loops of all the axes that are connected to the module. For a SERCOS interface module, the SERCOS ring also shuts down.
SERCOS Ring
Drive
Motor
Controller
Motion Module
SERCOS Ring
Drive
Motor

The controller automatically restarts the connections. The SERCOS ring also phases back up.

Inhibit only certain types of axes.

You can inhibit only these types of axes:

- AXIS_SERVO
- AXIS_SERVO_DRIVE
- AXIS_GENERIC_DRIVE

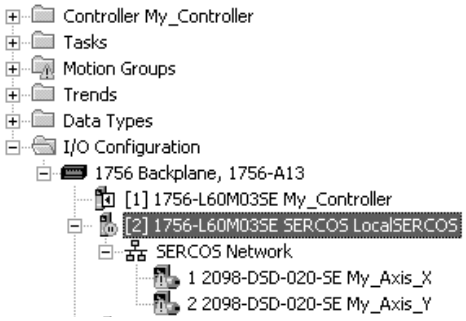
To inhibit all of the axes of a motion module, inhibit the module instead.

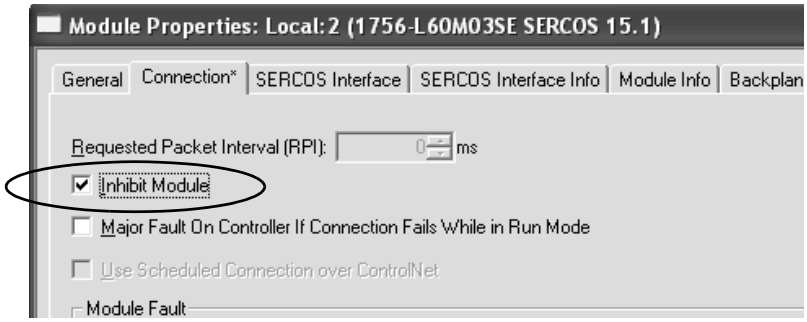
Do you want to inhibit all of the axes of a motion module?

- YES — Inhibit the motion module instead.
- NO — Inhibit the individual axes.

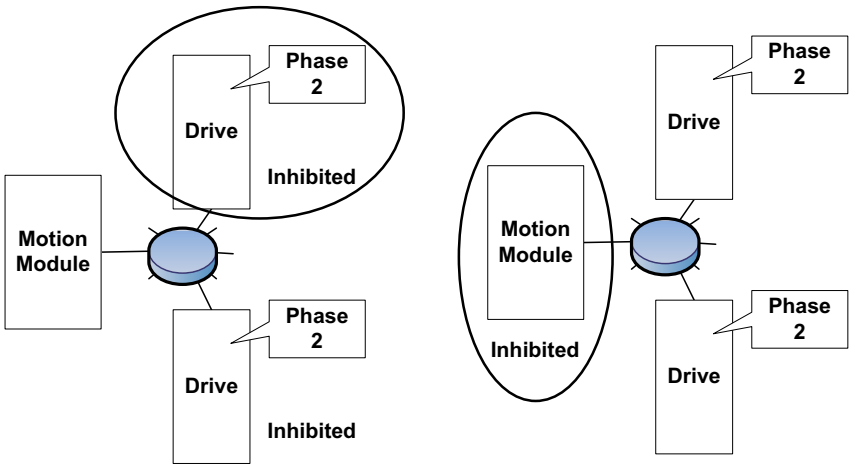
It's OK to inhibit all of the axes of a module one-by-one. It's just easier to inhibit the module.

Example: Suppose your motion module has two axes and you want to inhibit both of those axes. In that case, just inhibit the module.

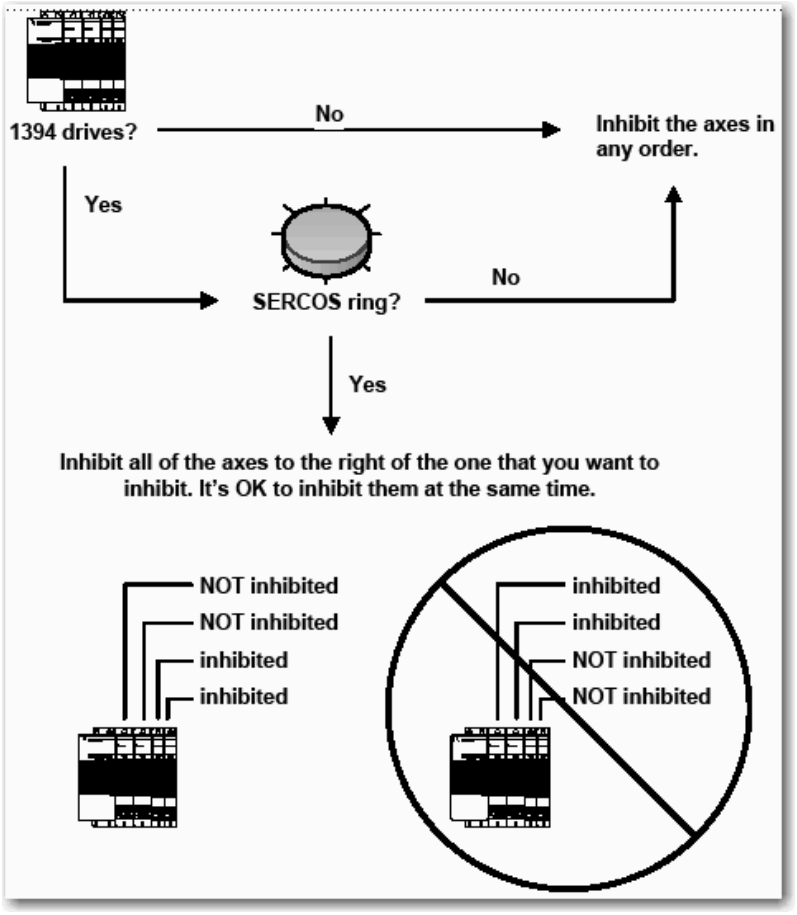




If you inhibit all of the axes on a SERCOS ring, the drives phase up to phase 2. This happens whether you inhibit all the axis individually or you inhibit the motion module.

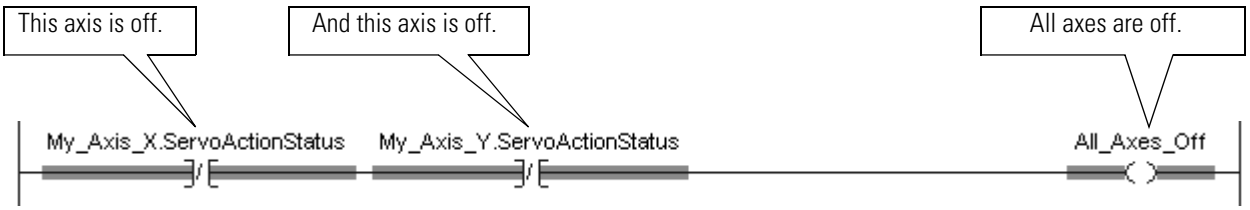


Do you have 1394 drives on a SERCOS ring?

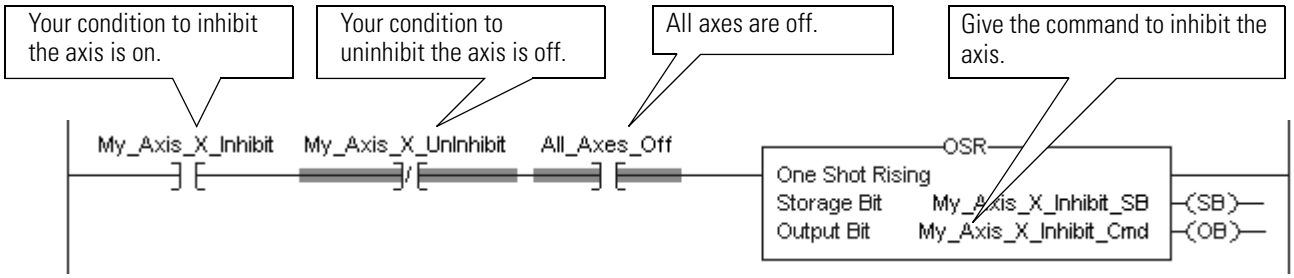


Example: Inhibit an Axis

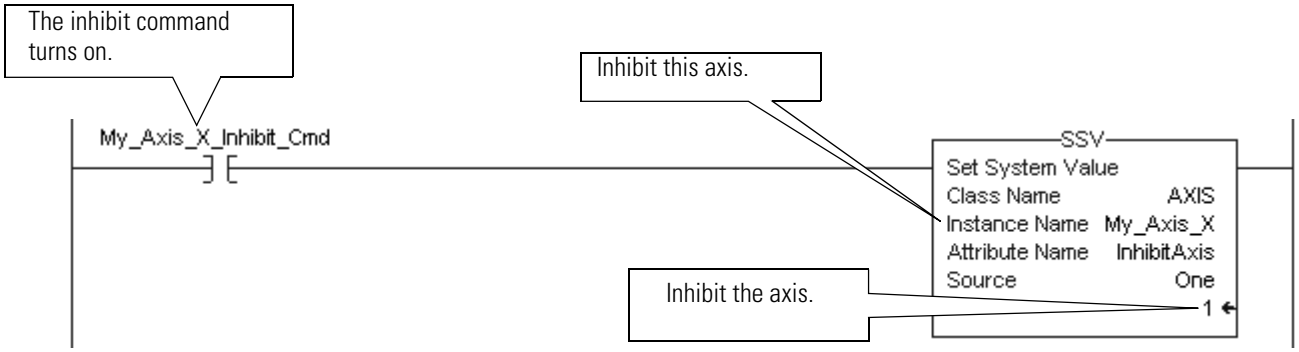
1. **Make sure all axes are off.**



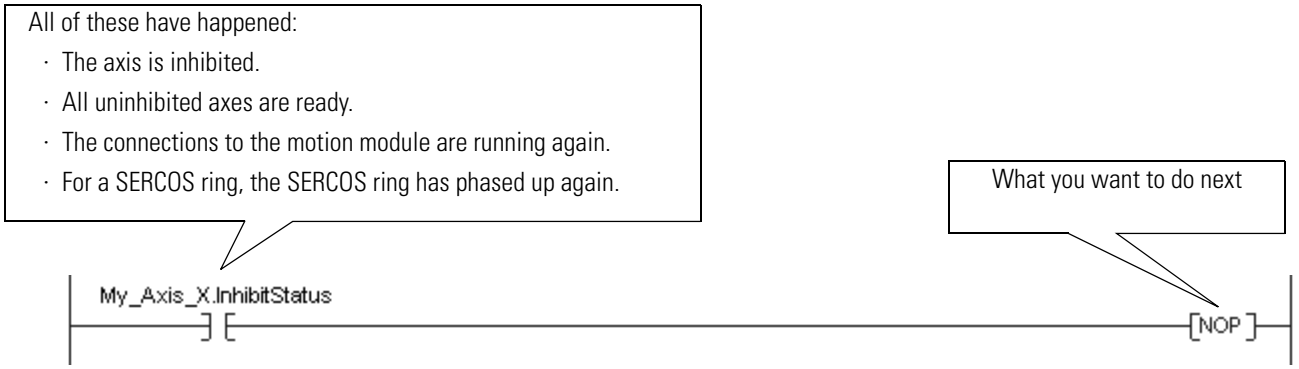
2. **Use a one-shot instruction to trigger the inhibit.**



3. **Inhibit the axis.**

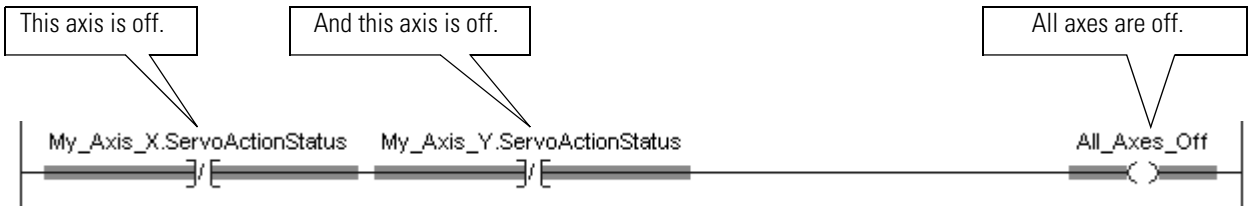


4. **Wait for the inhibit process to finish.**

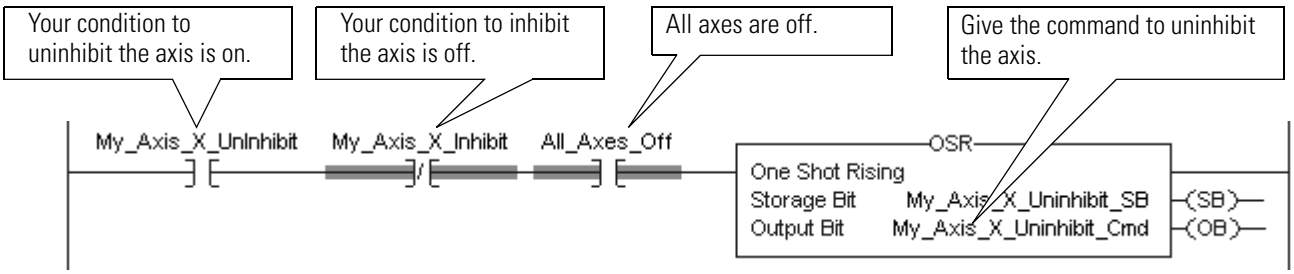

All of these have happened:
· The axis is inhibited.
· All uninhibited axes are ready.
· The connections to the motion module are running again.
· For a SERCOS ring, the SERCOS ring has phased up again.
What you want to do next
My_Axis_X.InhibitStatus
NOP

Example: Uninhibit an Axis

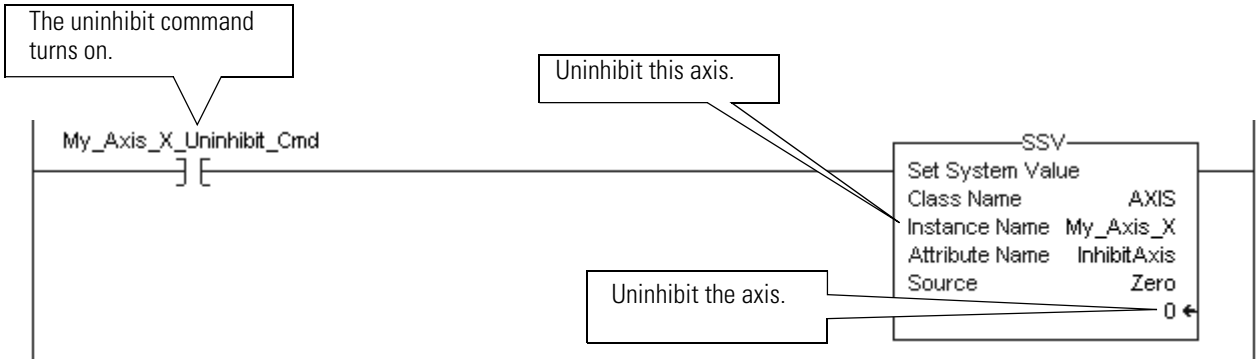
1. **Make sure all axes are off.**



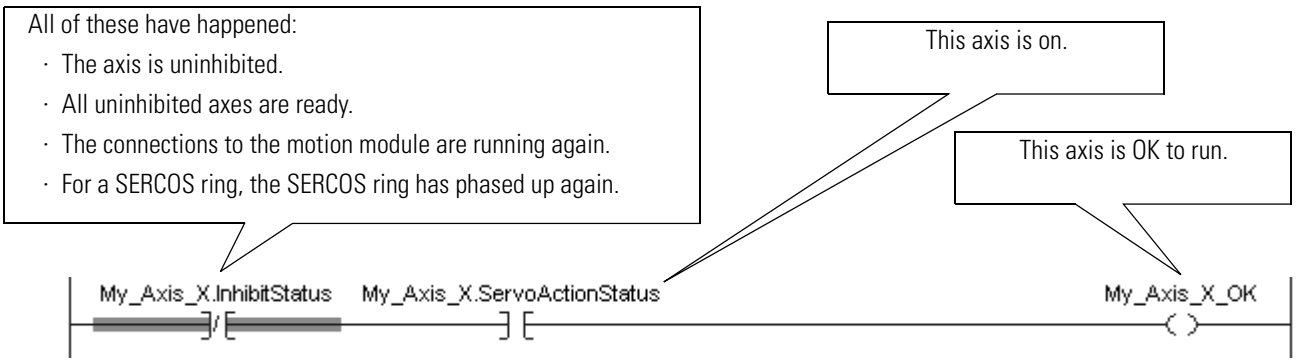
2. **Use a one-shot instruction to trigger the uninhibit.**



3. **Uninhibit the axis.**



4. **Wait for the inhibit process to finish.**


All of these have happened:
· The axis is uninhibited.
· All uninhibited axes are ready.
· The connections to the motion module are running again.
· For a SERCOS ring, the SERCOS ring has phased up again.
This axis is on.
This axis is OK to run.
My_Axis_X.InhibitStatus
My_Axis_X.ServoActionStatus
My_Axis_X_OK

Kinematics in RSLogix 5000 Software

Introduction

This chapter provides you with the information you need when using the Kinematics functionality within RSLogix 5000 software. This chapter also provides you with guidelines for robot-specific applications.

Controllers that Support Kinematics Functionality

These controllers support Kinematics functionality.

- ControlLogix 1756-L6x
- GuardLogix 1756-L6xS

Overview of Kinematics Functionality in RSLogix 5000 Software

RSLogix 5000 software provides built-in Kinematics transformation capability for controlling non-Cartesian robots. The Kinematics function provides seamless transformation of Cartesian coordinates to joint coordinates enabling the movement of rotating bases, elbows, and shoulders found in robotic arms. The benefits of RSLogix 5000 Kinematics integrated motion include:

- implementation of both sequential and robot-arm control within one controller.
- no synchronization or handshake code required to link the robot arm to the controller.
- all data is available via the controller resulting in one HMI for both the controller and robot arm.
- consistent hardware solution that reduces the need for spare parts and operator training.

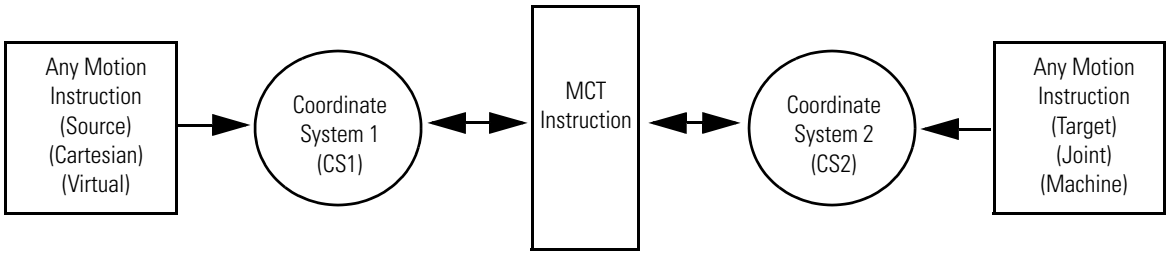
You program the RSLogix 5000 Kinematics function by using the standard motion instruction set in RSLogix 5000 software and these instructions:

- Motion Coordinated Transform (MCT)
- Motion Calculate Transform Position (MCTP)

IMPORTANT MCT and MCTP instructions are only supported on 1756-L6x and 1756-L6xS controllers.

The MCT instruction binds two coordinate systems and establishes a coordinate transformation between two coordinate systems. After the MCT

instruction is configured and executed you can program the robotic arm in the Cartesian coordinate system or the Joint coordinate system.


Any Motion Instruction (Source) (Cartesian) (Virtual)
Coordinate System 1 (CS1)
MCT Instruction
Coordinate System 2 (CS2)
Any Motion Instruction (Target) (Joint) (Machine)

The MCTP instruction is a calculate instruction that transforms a specified position from the source coordinate system into the target coordinate system and vice versa. This instruction can also be used for position recovery and teach position routines.

The RSLogix 5000 integrated Kinematics function provides you with an easy-to-use interface for:

- forward Kinematics - Joint coordinates are transformed to Cartesian coordinates.
- inverse Kinematics - Cartesian coordinates are transformed to Joint coordinates.

In RSLogix 5000 software Cartesian space is typically configured by using virtual axes and joint space is usually configured by using real axes.

Robot geometries supported for two and three axes are:

- Cartesian
- Articulated Dependant
- Articulated Independent
- Selective Compliant Assembly Robot Arm (SCARA) Independent
- Delta three-dimensional
- Delta two-dimensional
- SCARA Delta

Useful Terms

Understanding the terms used in this chapter enables you to properly configure your robot.

Term	Definition
Forward Kinematics	The solution of source positions given the target positions. In practice this would be computing the Cartesian positions given the Joint positions.
Forward Transform	The solution of source positions given target positions.
Inverse Kinematics	The solution of joint positions given Cartesian positions. Typically, converts Cartesian positions to joint positions.
Inverse Transform	The solution of target positions given source positions.
Joint axis	A rotary robotic coordinate axis typically having overtravel rather than rollover limits.
Kinematics	The family of mathematical equations that convert positions back and forth between two linked geometries.
Orientation	Robotic term for directional attitude or rotation about a point in Cartesian (3D) space. Orientation is expressed as three ordered rotations around the X, Y, and Z Cartesian axes.
Reference frame	An imaginary Cartesian coordinate system used to define a Cartesian origin and reference orientation.
Source system	One of two coordinate systems used in a Kinematics transform and having special properties. When connected to a target system by means of a Kinematics transform, motion commanded at the source system's inputs produces motion at both the source and target system's outputs (if the physical axes are connected).
Target system	One of two coordinate systems used in a Kinematics transform and having special properties. When connected to a source system by means of a Kinematics transform, motion commanded at the target system's inputs produces motion in both the source and target system's outputs (if the physical axes are connected).
Tool Center Point	All Kinematics programmed position (motion) is based on the Tool Center Point (TCP). To determine the TCP, you must enter information on the following RSLogix 5000 tabs: • Geometry - Enter values for Link Length (linear displacement), Zero Angle Orientation (angular rotation), and Base Offsets. These values, in combination with the selected Geometry type, defines the resulting Geometry's end-of-arm position. • Offsets - Enter value for End-effector offset; these are included when establishing the finalTCP position.
Transform	General term for conversion equations which map values in one coordinate space to values in another coordinate space.
Translation	Robotic term for a linear movement or offset in Cartesian (three-dimensional) space. Translation describes the distance between two Cartesian points.
Zero Angle Offset	Offset on a rotary axis in the Joint Coordinate system between where the Kinematics equations were derived and where you want your zero position to be.

Gather Information about Your Robot

Before you begin the configuration steps for the Kinematics transformation function you need to gather specific information about your robot and application parameters. Specifications for your robot can be found in the documentation provided by the manufacturer while other required information is application dependent. You should know this information before you begin configuring the RSLogix 5000 Kinematics for motion control.

· Robot geometry type

- Zero angle orientation
- Work envelope
- Link lengths
- Base offsets
- End-effector offsets
- Arm solution

Summary of Kinematic Steps

After you create a Joint (target) coordinate system tag for your Motion control project there are general steps to follow for Kinematics.

1. Determine and then configure the type of coordinate system you need for your robot.

   For help in determining your coordinate system type, refer to page 80.

2. Establish the Joint-to-Cartesian reference frame relationship.

   For more information regarding the joint-to-Cartesian reference frame, refer to the section about the type of robot you are using.

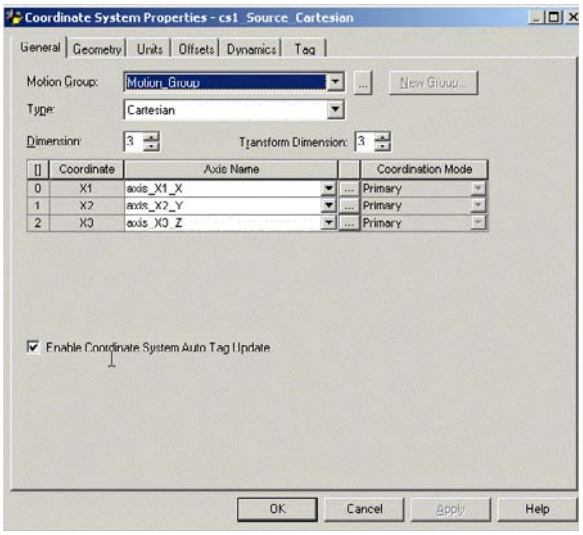
WARNING

The correct relationship between the Joint reference frame and the Cartesian reference frame must be established. Failure to do this can allow your robot to move to unexpected positions causing machine damage and/or injury or death to personnel.

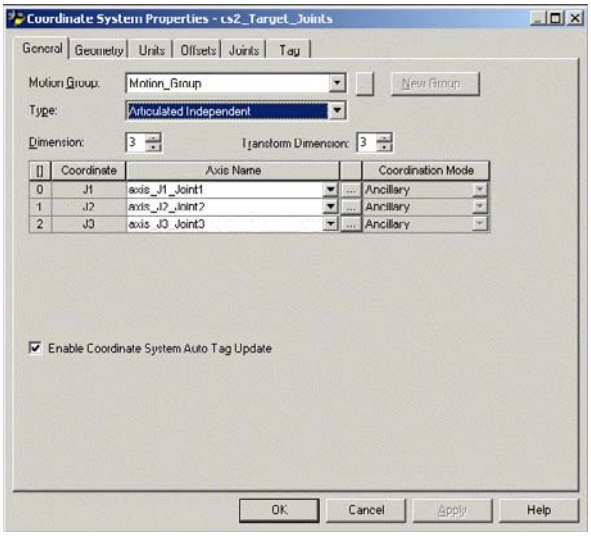
3. Calibrate your robot (if applicable).
4. Identify your robot work envelope.
5. Determine and then configure the following parameters:
   - link lengths
   - base offsets
   - end-effector offsets

6. Create the source and target coordinate systems.

Typical Cartesian Coordinate System Configuration for Articulated Independent robot.



Typical Joint Coordinate System Configuration for an Articulated Independent robot.



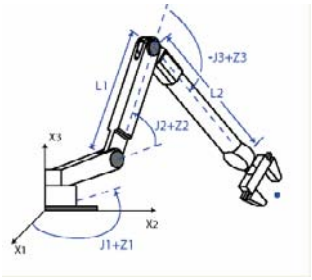
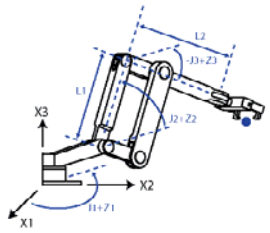
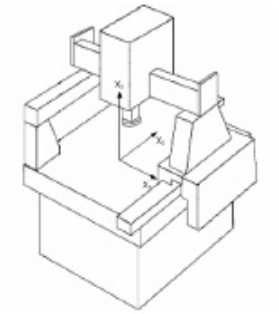
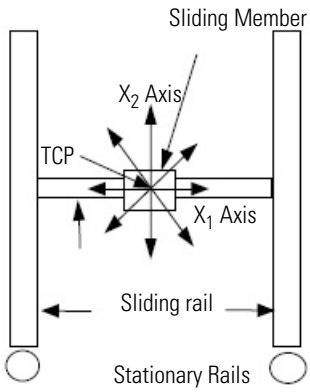
7. Save the project.

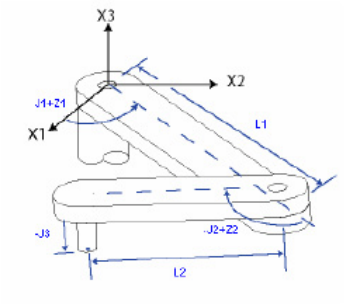
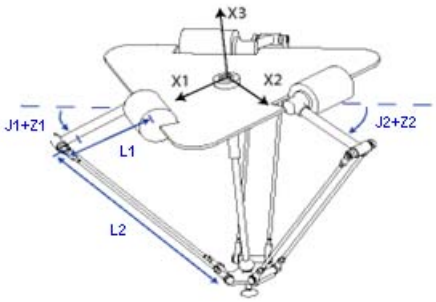
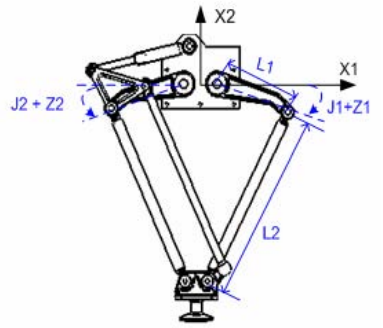
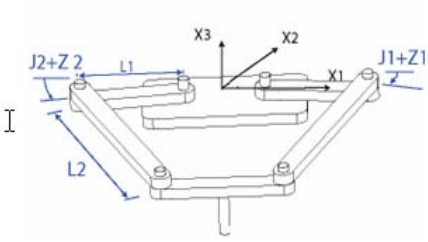
8. Download the Kinematic project to the 1756-L6xx controller and then, use the MCT instruction to link the Joint coordinate system to the Cartesian coordinate system.

   The Joint-to-Cartesian reference frame relationship is automatically established by the 1756-L6xx controller after the Joint coordinate system parameters (link lengths, base offsets, and end-effector offsets) are configured and the MCT instruction is enabled. For additional information about the MCT or MCTP instructions refer to the LOGIX5000 Controllers Motion Instructions, publication 1756-RM007.

For detailed steps about Creating and Configuring a Coordinate System, refer to Chapter 4, Create and Configure a Coordinate System.

Determine the Coordinate System Type

If your robot looks similar to	Your Coordinate System type is
	Articulated Independent For configuration information go to page 82.
	Articulated Dependent For configuration information go to page 92.
	Cartesian This illustration shows a typical Gantry machine. For configuration information go to page 101.
	Cartesian This illustration shows a typical H-bot. For configuration information go to page 102.

If your robot looks similar to	Your Coordinate System type is
	SCARA Independent For configuration information go to page 104.
	Three-dimensional Delta For configuration information go to page 108.
	Two-dimensional Delta For configuration information go to page 117.
	SCARA Delta For configuration information go to page 122.

Configure an Articulated Independent Robot

Use these guidelines when configuring an Articulated Independent robot.

WARNING

Before turning ON the Transform and/or establishing the reference frame, be sure to do the following for the joints of the target coordinate system:

- Set and enable the soft travel limits.
- Enable the hard travel limits.

Failure to do this can allow the robot to move outside of the work envelope causing machine damage and/or serious injury or death to personnel.

Establish the Reference Frame for an Articulated Independent Robot

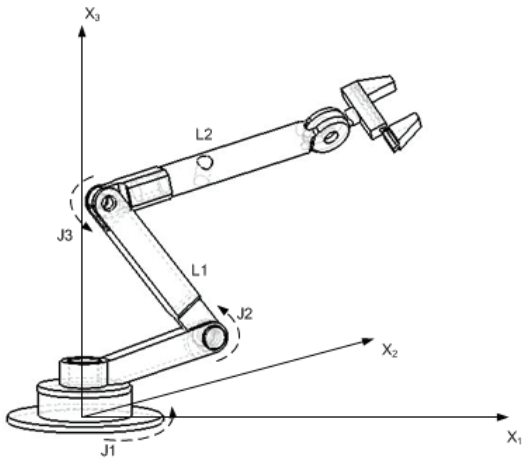
The reference frame is the Cartesian coordinate frame that defines the origin and the three primary axes (X1, X2, and X3). These axes are used to measure the real Cartesian positions.

WARNING

Failure to properly establish the correct reference frame for your robot can cause the robotic arm to move to unexpected positions causing machine damage and/or injury or death to personnel.

The reference frame for an Articulated Independent robot is located at the base of the robot as shown in the figure below.

Figure 1 - Articulated Independent

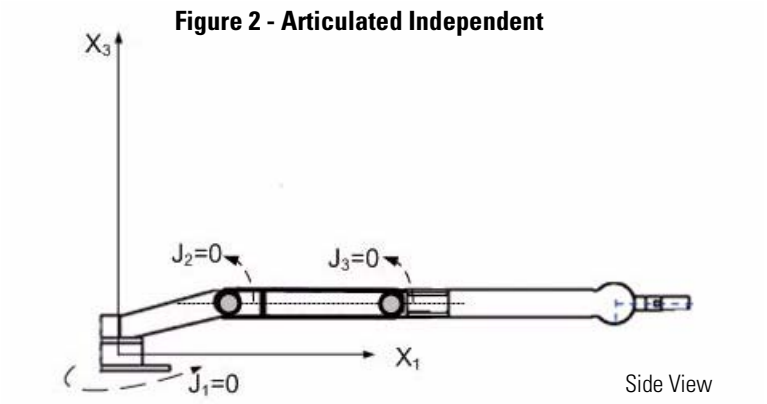


Before you begin establishing the Joint-to-Cartesian reference frame relationship, it is important to know some information about the Kinematic mathe-

matical equations used in the ControlLogix 1756-L6xx controllers. The equations were written as if the Articulated Independent robot joints were positioned as shown in Figure 2 - Articulated Independent.

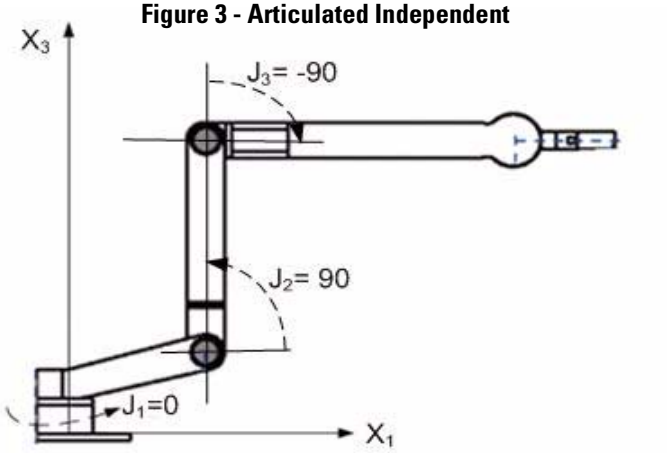
- +J1 is measured counterclockwise around the +X3 axis starting at an angle of J1=0 when L1 and L2 are both in the X1-X2 plane.
- +J2 is measured counterclockwise starting with J2=0 when L1 is parallel to X1-X2 plane.
- +J3 is measured counterclockwise with J3=0 when L2 is aligned with link L1.

Figure 2 - Articulated Independent



When your robot is physically in the position illustrated in	The RSLogix 5000 Actual Position tags for the axes must read as
Figure 2 - Articulated Independent	J1 = 0 J2 = 0 J3 = 0
Figure 3 - Articulated Independent	J1 = 0 J2 = 90 J3 = -90

Figure 3 - Articulated Independent



Side View

If your robot's physical position **and** joint angle values cannot match those shown in Figure 2 - Articulated Independent or Figure 3 - Articulated Independent, then use one of the Alternate Methods for Establishing the Joint-to-Cartesian reference frame relationship.

Alternate Methods for Establishing the Reference Frame for an Articulated Independent Robot

The following methods let you establish a reference frame for an Articulated Independent robot.

For each	Use one of these methods to establish the reference frame
Incremental axis	Each time the robot's power is cycled.
Absolute axis	Only when you establish absolute home.

- Method 1 - establishes a Zero Angle Orientation and allows the configured travel limits and home position on the joint axes to remain operational. Use this method if you are operating the axes between the travel limits determined prior to programming a Motion Redefine Position (MRP) instruction and want these travel limits to stay operational.
- Method 2 - uses a MRP instruction to redefine the axes position to align with the Joint reference frame. This method may require the soft travel limits to be adjusted to the new reference frame.

Method 1 - Establishing a Reference Frame

Each axis for the robot has the mechanical hard stop in each of the positive and negative directions. Manually move or press each axes of the robot against its associated mechanical hard stop and redefine it to the hard limit actual position provided by the robot manufacturer. J1 is the axis at the base of the robot that rotates around X3.

When the robot is moved so that Link1 is parallel to the X3 axis and Link2 is parallel to X1 axis as shown in Figure 2 - Articulated Independent, the RSLogix 5000 Actual Position tag values should be equal to:

- J1 = 0
- J2 = 90°
- J3 = -90°

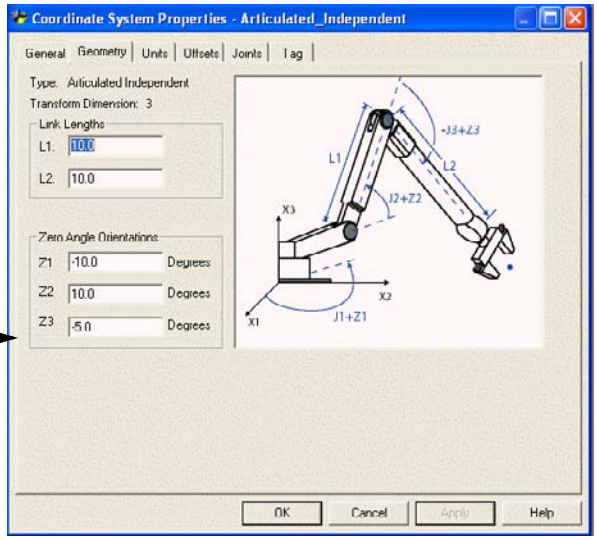
If the RSLogix 5000 Positions tags do not correspond to these values, configure the Zero Angle Orientation for the joint or joints that do not correspond.

If the RSLogix 5000 software read-out values are	Set the Zero Angle Orientations on the Coordinate System Properties dialog to
J1 = 10	Z1 = -10
J2 = 80	Z2 = 10
J3 = -85	Z3 = -5

The Joint-to-Cartesian reference frame relationship is automatically established by the 1756-L6xx controller after the Joint coordinate system parameters (link lengths, base offsets, and end-effector offsets) are configured and the MCT instruction is enabled.

Setting the Zero Angle Orientations

Set the Zero Angle Orientations.



Method 2 - Establishing a Reference Frame

Position the robot so that:

- Link1 is parallel to the X3 axis.
- Link2 is parallel to X1 axis.

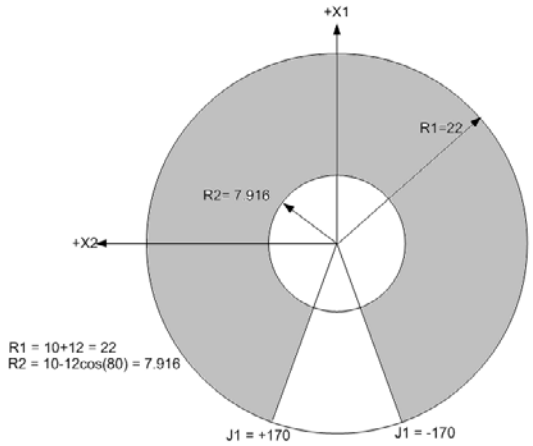
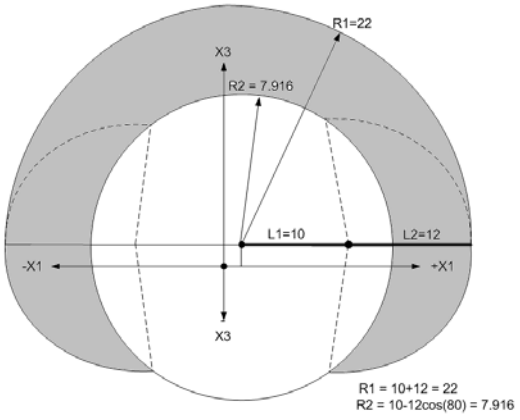
Program a MRP instruction for all three axes with the following values:

- J1 = 0
- J2 = 90°
- J3 = -90°

The Joint-to-Cartesian reference frame relationship is automatically established by the 1756-L6xx controller after the Joint coordinate system parameters (link lengths, base offsets, and end-effector offsets) are configured and the MCT instruction is enabled.

Identify the Work Envelope for an Articulated Independent Robot

The work envelope is the three-dimensional region of space that defines the reaching boundaries for the robot arm. The work envelope for an articulated robot is ideally a complete sphere having an inner radius equal to L1- L2 and outer radius equal to L1+L2. Due to the range of motion limitations on individual joints, the work envelope may not be a complete sphere.

If the range-of-motion values for the articulated robot are	Typically, the work envelope would be
J1 = ± 170 J2 = 0 to 180 J3 = ± 100 L1= 10 L2 = 12	 Top view - Depicts the envelope of the tool center point sweep in J1 and J3 while J2 remains at a fixed position of 0°.  Side view - Depicts the envelope of the tool center point sweep in J2 and J3 while J1 remains at a fixed position of 0°.

Define the Configuration Parameters for an Articulated Independent Robot

RSLogix 5000 software can be configured for control of robots with varying reach and payload capacities. As a result, it is very important to know the configuration parameter values for your robot including:

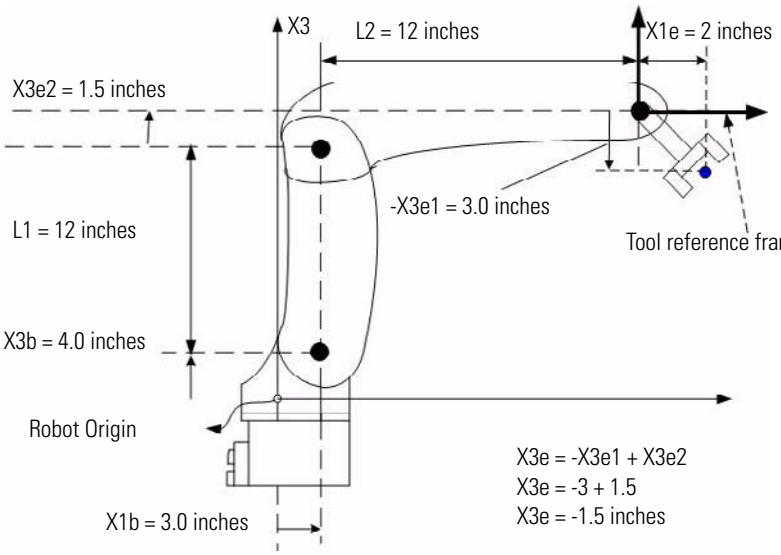
- link lengths
- base offsets
- end-effector offsets

The configuration parameter information is available from the robot manufacturer.

IMPORTANT Be sure that the values for the link lengths, base offsets and end-effector offsets are entered into the Configuration Parameters dialog using the same measurement units.

The following example illustrates the typical configuration parameters for an Articulated Independent robot.

Figure 4 - Articulated Independent



If the robot is two-dimensional, then X3b and X3e would be X2b and X2e respectively.

Link Lengths

Link lengths are the rigid mechanical bodies attached at joints.

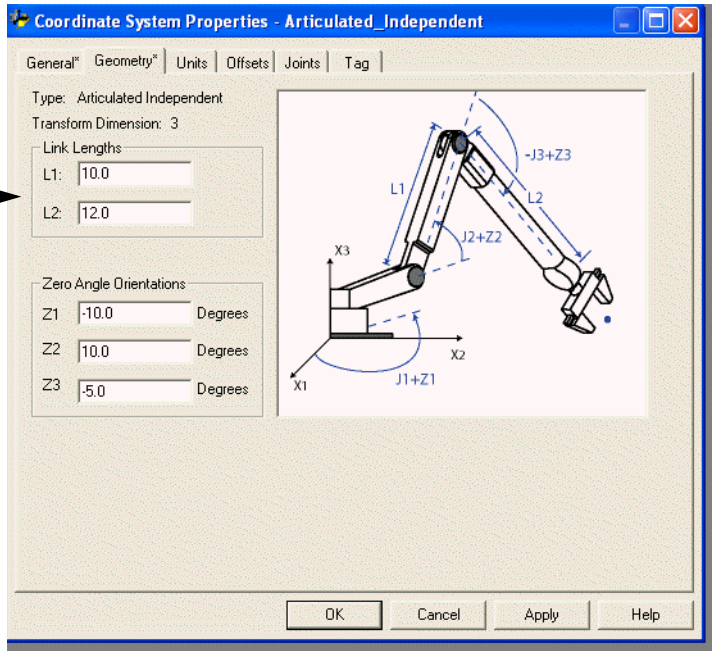
For an articulated independent robot with	The length of	Is equal to the value of the distance between
2 dimensions	L1	J1 and J2
	L2	J2 and the end-effector
3 dimensions	L1	J2 and J3
	L2	J3 and the end-effector

Example of Link Lengths for an Articulated Independent Robot

Enter the Link Length values.

For the robot shown in Figure 4 - Articulated Independent, the Link Length values are:

- L1 = 10.0
- L2 = 12.0

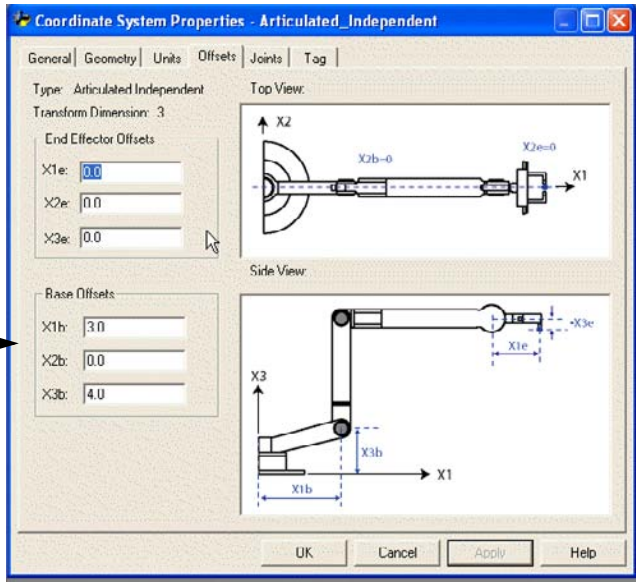


Base Offsets

The base offset is a set of coordinate values the redefines the origin of the robot. The correct base offset values are typically available from the robot manufacturer. Enter the values for the base offsets in the X1b and X3b fields of the Coordinate System Properties dialog.

Example of Base Offsets for an Articulated Independent Robot

Enter the Base Offset values.



For the robot shown in our example, the Base Offset values are:

- X1b = 3.0
- X3b = 4.0

End-effector Offsets

The robot can have an end-effector attached to the end of robot link L2. If there is an attached end-effector then you must configure the end-effector offset value on the Coordinate System Properties dialog. The end-effector offsets are defined with respect to the tool reference frame at the tool tip.

Some robots also have an offset defined for the J3 joint as illustrated in the robot example (refer to Figure 4 - Articulated Independent on page 88). You can account for this value when computing the X3e end-effector offset value. In

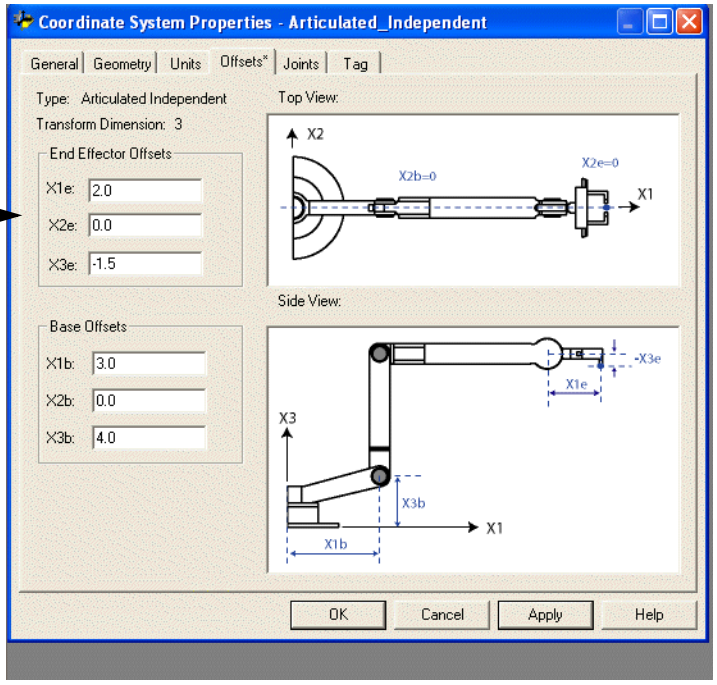
Figure 4 - Articulated Independent the value for X3e offset is entered as the sum of X3e1+X3e2 (-3+1.5 = -1.5). The configured value for X3e is -1.5.

Example of End-effectors for an Articulated Independent Robot

Enter the end-effector offset values.

For the robot shown in our example, the end-effector values are:

- X1e = 2.0
- X3e = -1.5



Configure an Articulated Dependent Robot

The Articulated Dependent robot has motors for the elbow and the shoulder located at the base of the robot. The dependent link controls J3 at the elbow. Use these guidelines when configuring an Articulated Dependent robot.

WARNING

Before turning ON the Transform and/or establishing the reference frame, be sure to do the following for the joints of the target coordinate system:

- Set and enable the soft travel limits.
- Enable the hard travel limits.

Failure to do this can allow the robot to move outside of the work envelope causing machine damage and/or serious injury or death to personnel.

Establish the Reference Frame for an Articulated Dependent Robot

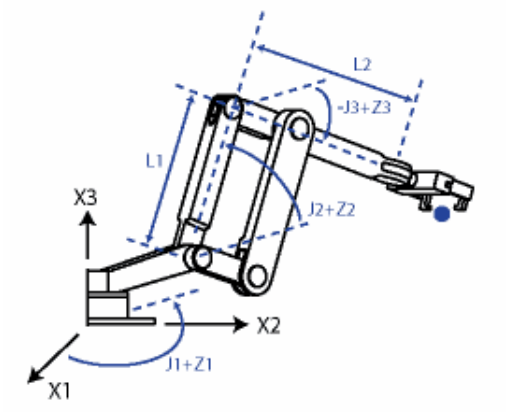
The reference frame is the Cartesian (typically the source) coordinate frame that defines the origin and the three primary axes (X1, X2 and X3). These are used to measure the real Cartesian positions.

WARNING

Failure to properly establish the correct reference frame for your robot can cause the robotic arm to move to unexpected positions causing machine damage and/or injury or death to personnel.

The reference frame for an Articulated Dependent robot is at the base of the robot as shown in the figure below.

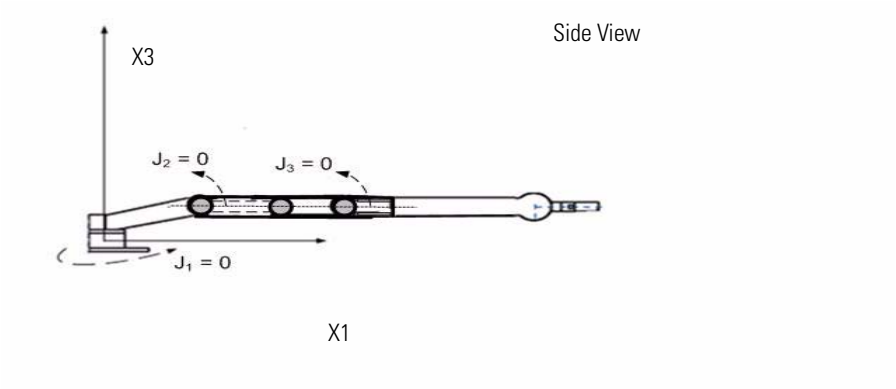
Figure 1 - Articulated Dependent



Before you begin establishing the Joint-to-Cartesian reference frame relationship, it is important to know some information about how the Kinematic mathematical equations in the ControlLogix 1756-L6xx controllers were written. The equations were written as if the Articulated Dependent robot joints were positioned as shown in Figure 2 - Articulated Dependent.

- +J1 is measured counterclockwise around the +X3 axis starting at an angle of J1=0 when L1 and L2 are both in the X1-X2 plane.
- +J2 is measured counterclockwise starting with J2=0 when L1 is parallel to X1-X2 plane.
- +J3 is measured counterclockwise with J3=0 when L2 is parallel to the X1-X2 plane.

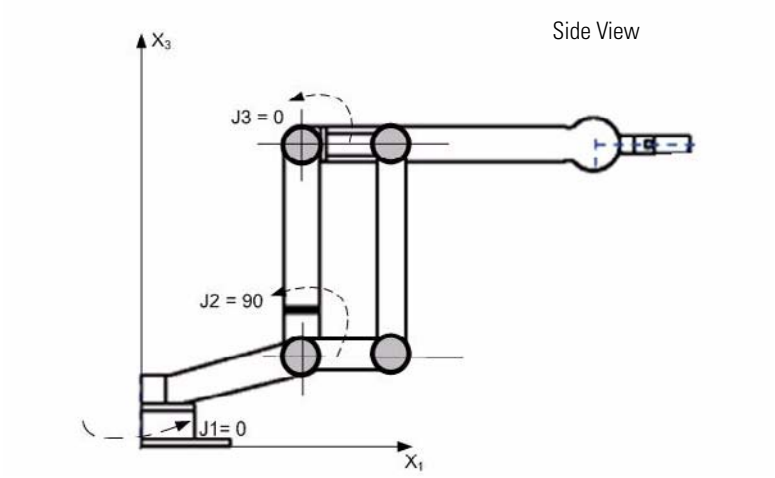
Figure 2 - Articulated Dependent



When your robot is physically in the position illustrated in	The RSLogix 5000 Actual Position tags for the axes must read as
Figure 2 - Articulated Dependent	J1 = 0 J2 = 0 J3 = 0
Figure 3 - Articulated Dependent	J1 = 0 J2 = 90 J3 = 0

.

Figure 3 - Articulated Dependent



If your robot's physical position **and** joint angle values cannot match those shown in Figure 2 - Articulated Dependent or in Figure 3 - Articulated Dependent then, use one of the methods outlined in this section to establish the Joint-to-Cartesian reference frame relationship.

WARNING

Failure to properly establish the correct reference frame for your robot can cause the robotic arm to move to unexpected positions potentially resulting in damage to property or injury to personnel.

Alternate Methods for Establishing the Reference Frame for an Articulated Dependent Robot

The following methods let you establish a reference frame for an Articulated Independent robot.

For each	Use one of these methods to establish the reference frame
Incremental axis	Each time the robot's power is cycled.
Absolute axis	Only when you establish absolute home.

- Method 1 - establishes a Zero Angle Orientation and allows the configured travel limits and home position on the joint axes to remain operational. Use this method if you are operating the axes between the travel limits determined prior to programming a Motion Redefine Position (MRP) instruction and want these travel limits to stay operational.

- Method 2 - uses a Motion Redefine Position (MRP) instruction to redefine the axes position to align with the Joint reference frame. This method may require the soft travel limits to be adjusted to the new reference frame.

Method 1 - Establishing a Reference Frame

Each axis for the robot has the mechanical hard stop in each of the positive and negative directions. Manually move or press each axes of the robot against its associated mechanical hard stop and redefine it to the hard limit actual position provided by the robot manufacturer. J1 is the axis at the base of the robot that rotates around X3.

When the robot is moved so that Link1 is parallel to the X3 axis and Link2 is parallel to X1 axis as shown in Figure 3 - Articulated Dependent, the RSLogix5000 values for the Actual Position tags should be:

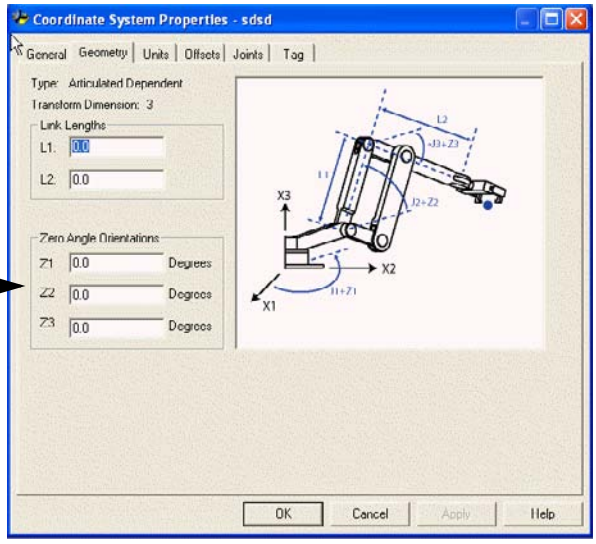
- J1 = 0
- J2 = 90°
- J3 = 0°

If the RSLogix 5000 Actual Position tags do not show these values, configure the Zero Angle Orientation for the joint or joints that do not correspond.

If the RSLogix 5000 software read-out values are	Set the Zero Angle Orientations on the Coordinate System Properties dialog to
J1 = 10 J2 = 80 J3 = 5	Z1 = -10 Z2 = 10 Z3 = -5

Example of Zero Angle Orientation for an Articulated Dependent Robot

Set the Zero Angle Orientations.



Method 2 - Establishing a Reference Frame

Position the robot so that:

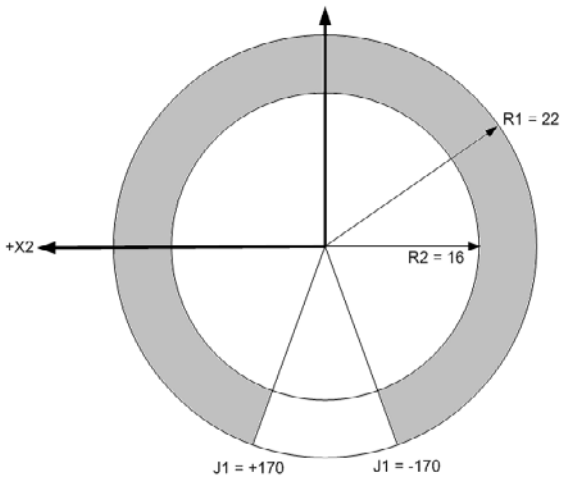
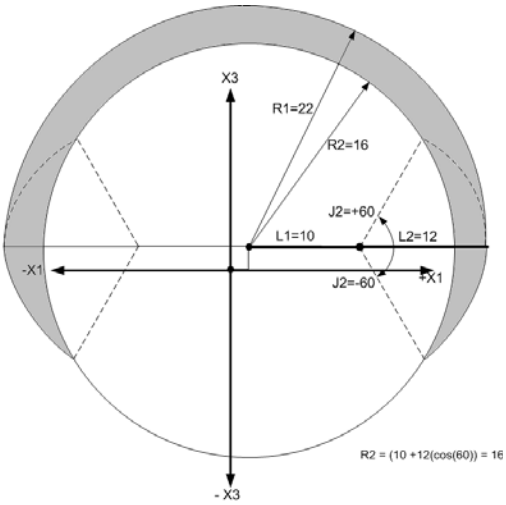
- L1 is parallel to the X3 axis.
- L2 is parallel to X1 axis.

Program a Motion Redefine Position (MRP) instruction for all the three axis to with the following values 0, 90, and 0° .

The Joint-to-Cartesian reference frame relationship is automatically established by the 1756-L6xx controller after the Joint coordinate system parameters (link lengths, base offsets, and end-effector offsets) are configured and the MCT instruction is enabled.

Identify the Work Envelope for an Articulated Dependent Robot

The work envelope is the three-dimensional region of space defining the reaching boundaries for the robot arm. The work envelope of an articulated robot is ideally a complete sphere having an inner radius equal to |L1- L2| and outer radius equal to |L1+L2|. However, due to the range of motion limitations on individual joints, the work envelope may not be a complete sphere.

If the range-of-motion values for the articulated robot are	Typically, the work envelope would be
J1 = ± 170 J2 = 0 to 180 J3 = ± 60 L1= 10 L2 =12	 Top view - Depicts the envelope of the tool center point sweep in J1 and J3 while J2 remains at a fixed position of 0°.  Side view - Depicts the envelope of the tool center point sweep in J2 and J3 while J1 remains at a fixed position of 0°.

Define Configuration Parameters for an Articulated Dependent Robot

RSLogix 5000 software can be configured for control of robots with varying reach and payload capacities. As a result, it is very important to know the configuration parameter values for your robot including:

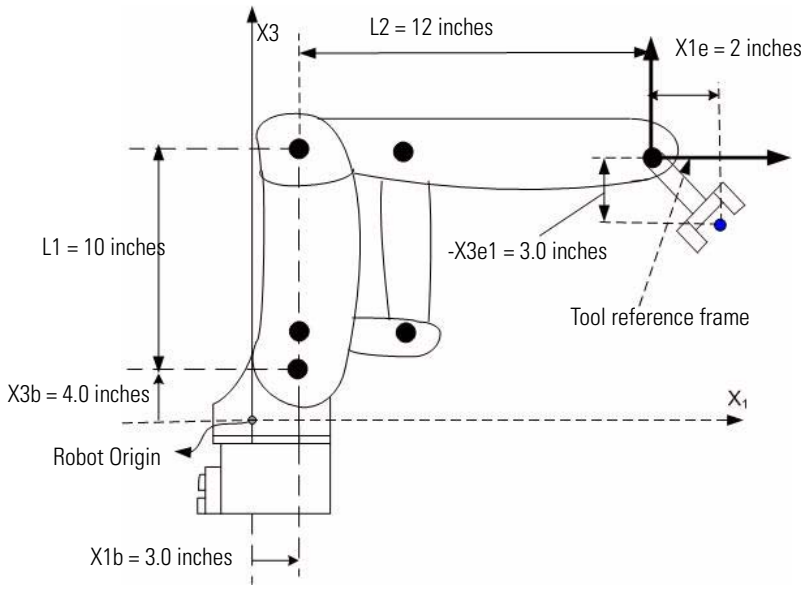
- link lengths
- base offsets
- end-effector offsets

The configuration parameter information is available from the robot manufacturer.

IMPORTANT Be sure that the values for the link lengths, base offsets and end-effector offsets are entered into the Configuration Parameters dialog using the same measurement units.

The following example illustrates the typical configuration parameters for an Articulated Dependent robot.

Figure 4 - Articulated Dependent



If the robot is two-dimensional, then X3b and X3e would be X2b and X2e respectively.

Link Lengths

Link lengths are the rigid mechanical bodies attached at joints.

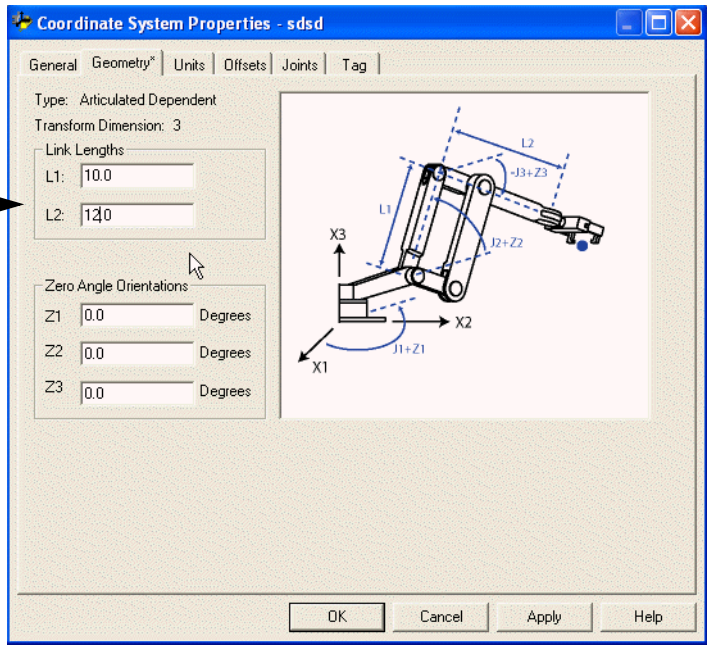
For an articulated dependent robot with	The length of	Is equal to the value of the distance between
Two-dimensions	L1	J1 and J2
	L2	J2 and the end-effector
Three-dimensions	L1	J2 and J3
	L2	J3 and the end-effector

Example of Link Lengths for an Articulated Dependent Robot

Enter the Link Length values.

For the robot shown in our example, the Link Length values are:

- L1 = 10.0
- L2 = 12.0



Base Offsets

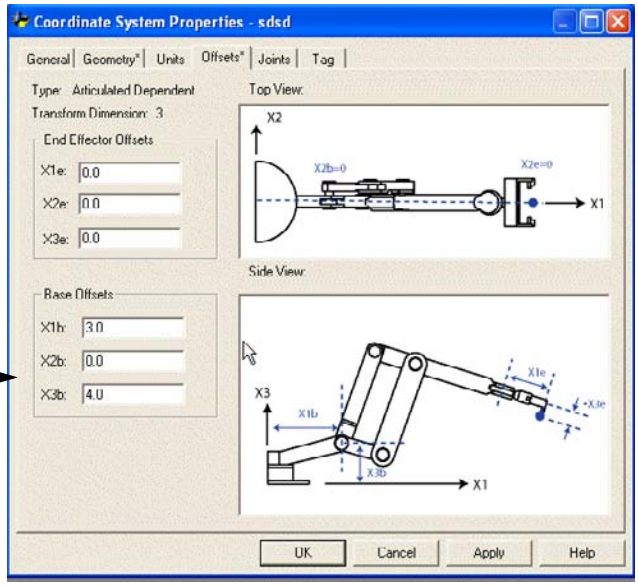
The base offset is a set of coordinate values the redefines the origin of the robot. The correct base-offset values are typically available from the robot manufacturer. Enter the values for the base offsets in the X1b and X3b fields of the Coordinate System Properties dialog.

Example of Base Offsets for an Articulated Independent Robot

Enter the Base Offset values.

For the robot shown in our example, the Base Offset values are:

- X1b = 3.0
- X3b = 4.0



End-effector Offsets

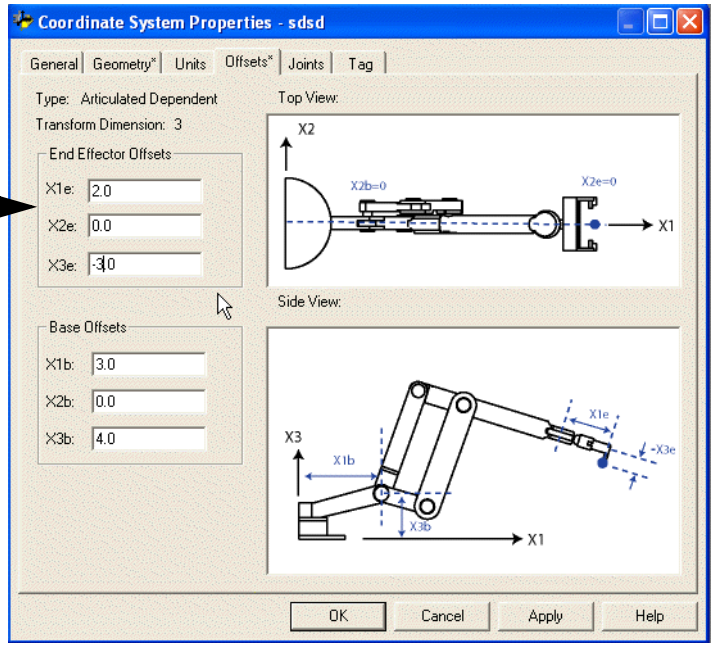
The robot can have an end-effector attached to the end of robot link L2. If there is an attached end-effector then you must configure the end-effector offset value on the Coordinate System Properties dialog. The end-effector offsets are defined with respect to the tool reference frame at the tool tip.

Example of End-effector Values for an Articulated Independent Robot

Enter the end-effector offset values.

For the robot shown in our example, the end-effector values are:

- X1e = 2.0
- X3e = -3.0



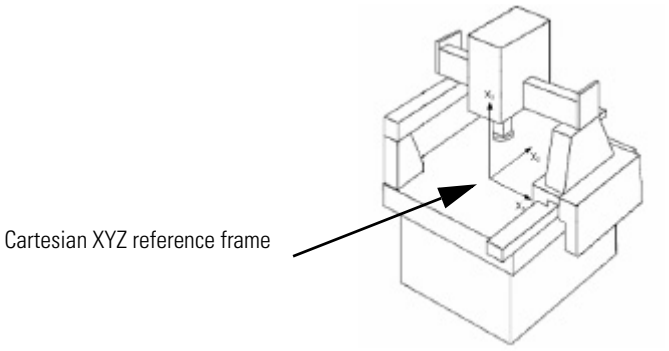
Configure a Cartesian Gantry Robot

Use these guidelines when configuring a Cartesian Gantry robot.

Establish the Reference Frame for a Cartesian Gantry Robot

For a Cartesian Gantry robot the reference frame is an orthogonal set of X1, X2, and X3 axes positioned anywhere on the Cartesian robot. All global coordinate measurements (points) are relative to this reference frame. Typically, the reference frame is aligned with the X1, X2, and X3 axes of the machine

Cartesian Reference Frame.



To establish a Local coordinate system with axes positions different from the reference frame, use the Motion Redefine Position (MRP) instruction to reset the position register. You can also use the Offset Vector in the MCT transform instruction to establish an offset between the Local coordinate system and the reference frame.

For more information about Motion Instructions, refer to the Logix5000 Controllers Motion Instruction Reference manual, publication number 1756-RM007.

Identify the Work Envelope for a Cartesian Gantry Robot

The work envelope for a Cartesian Gantry robot is typically a solid rectangle of length, width and height that is equal to the axis travel limits.

Define Configuration Parameters for a Cartesian Gantry Robot

Link Lengths

Does not apply to a Cartesian Gantry robot configuration.

Base Offsets

Does not apply to a Cartesian Gantry robot configuration.

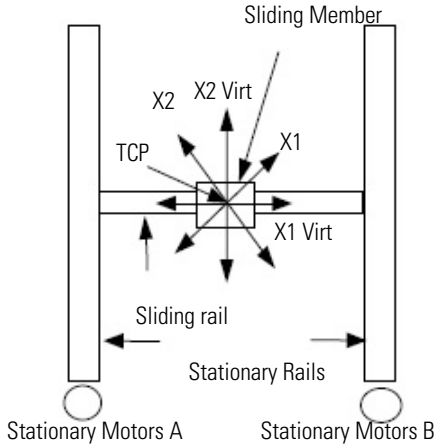
End-effector Offsets

Does not apply to a Cartesian Gantry robot configuration.

Configure a Cartesian H-bot

The H-bot is a special type of Cartesian two-axis gantry robot. This type of machine has three rails positioned in the form of a letter H. Two motors are positioned at the end of each leg of the robot. Unlike a standard gantry robot, neither motor is riding on top of the moving rails. Use these guidelines when configuring a Cartesian H-bot.

Cartesian H-bot



In the Cartesian H-bot illustration above, the X1 and X2 axes are the real axes on the robot. X1 Virt and X2 Virt are configured as the virtual axes.

The configuration of the H-bot mechanical linkages enable it to move at a 45° angle to the axes when either motor A or motor B is rotated.

For example, when:

- Motor A (X1 axis) is rotated, the robot move along a straight line at + 45° angle.
- Motor B (X2 axis) is rotated the machine moves at an angle of -45° .
- Motors A and B are both rotated clockwise at the same speed then, the machine moves along a horizontal line.

- Motors A and B are both rotated counterclockwise at the same speed then, the machine moves along a vertical line.

Any X,Y position can be reached by properly programming the two motors.

For example, a move of (X1 = 10, X2 = 0) causes the X1X2 axes to move to a position of (X1=7.0711, X2=7.0711). A move to (X1=10, X2 =10) causes the robot to move to a position of (X1=0, X2=14.142).

While this configuration might be very confusing for a programmer, utilizing the RSLogix 5000 software Kinematics function configured with two Cartesian coordinate systems and a -45° rotation easily performs the function.

To configure two Cartesian coordinate systems, Coordinate system 1 (CS1) and Coordinate system 2 (CS2), each containing two linear axes, use the following steps.

1. Configure CS1 to contain the virtual X1 and X2 axes.
2. Configure CS2 to contain the real X1 and X2 axes.
3. Configure the Orientation vector of the MCT instruction as (0,0, -45) a negative degree rotation around the X3 axis.
4. Configure the Translation vector as (0, 0, 0).
5. Link the CS1 and CS2 using a MCT instruction.
6. Home the H-bot and then program all moves in CS1.

The machine moves the tool center point (TCP) to the programmed coordinates in CS2. The -45° rotation introduced by the Kinematics, counteracts the 45° rotation introduced by the mechanics of the machine and the H-bot moves to the CS1 configured coordinates. As a result, a programmed move of X1virt=10, X2virt=5 moves to a real mechanical position of X1=10, X2=5.

Establish the Reference Frame for a Cartesian H-bot

For a Cartesian H-bot the Base coordinate system is an orthogonal set of X1, X2 axes postponed anywhere on the Cartesian H-bot. The angular rotation of the reference frame may not be rotated for this robot since the angular rotation vector is used to achieve the 45° rotation required for the mechanical operation.

Identify the Work Envelope for a Cartesian H-bot

The work envelope for a Cartesian H-bot is a rectangle of length and width equal to the axis soft travel limits.

Define Configuration Parameters for a Cartesian H-bot

Link Lengths

Does not apply to a Cartesian H-bot configuration.

Base Offsets

Does not apply to a Cartesian H-bot configuration.

End-effector Offsets

Does not apply to a Cartesian H-bot configuration.

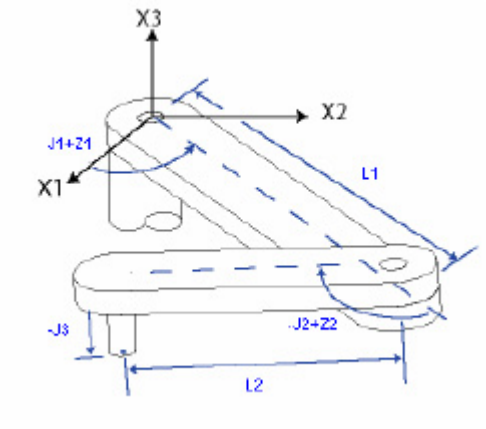
Configure a SCARA Independent

The typical SCARA Independent robot has two revolute joints and a single prismatic joint. This robot is identical to the Articulated Independent two-dimensional robot except that the X1-X2 plane is tilted horizontally with a third linear axis in the vertical direction. Use these guidelines when configuring a SCARA Independent robot.

Establish the Reference Frame for a SCARA Independent Robot

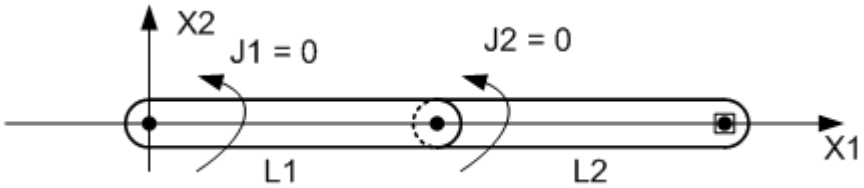
The reference frame for the SCARA Independent geometry is at the base of link L1.

SCARA Independent Robot Reference Frame



The internal Kinematic equations are written as if the start position for the SCARA Independent robot joints are as shown in this illustration.

Joint and Link Start Position that Kinematics Equations use for the SCARA Independent Robots



Top View

- +J1 is measured counterclockwise around +X3 axis starting at an angle of J1 =0.0 when L1 is along the X1 axis.
- +J2 is measured counterclockwise starting with J2 = 0 when Link L2 is aligned with Link L1.
- +J3 is a prismatic axis that moves parallel to +X3 axis.

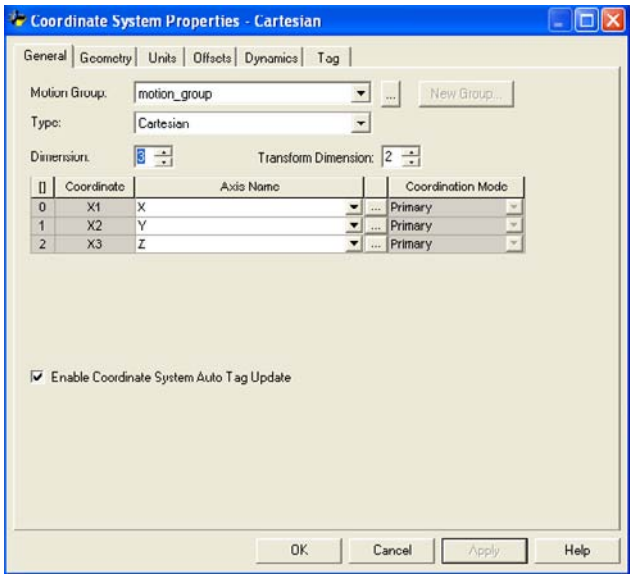
For information about alternate methods for establishing a reference frame see the Configure an Articulated Independent Robot section on page 82.

When configuring the parameters for the Source coordinate system and the Target coordinate system for a SCARA Independent robot, keep the following information in mind:

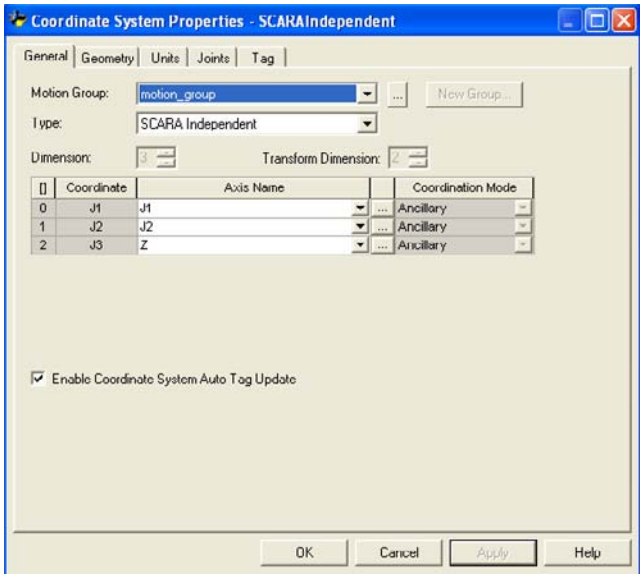
- The transform dimension value should be set to two for both the source and target coordinate systems because only J1 and J2 are involved in the transformations.
- The Z axis is configured as a member of both the source and target coordinate systems.

For additional information about establishing a reference frame, refer to the section entitled Configure an Articulated Independent Robot in this manual.

Example Source and Target Coordinate Systems for a SCARA Independent Robot



Source Coordinate System Configuration



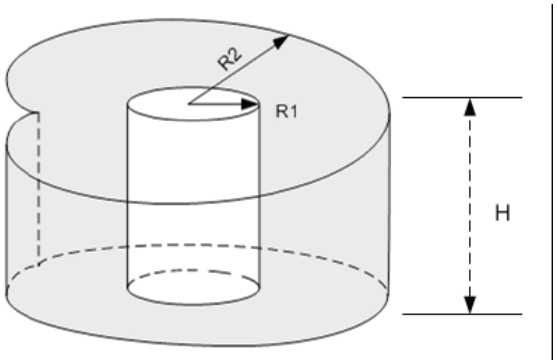
Target Coordinate System Configuration

Identify the Work Envelope for a SCARA Independent Robot

The work envelope is the three-dimensional region of space that defines the reaching boundaries for the robot arm. The work envelope for the SCARA Independent robot should be a hollow cylinder with:

- a height equal to the travel limit of the J3 axis.
- an inner radius (R1) equal to |L1-L2|.
- an outer radius (R2) equal to |L1+L2|.

Example Work Envelope for a SCARA Independent Robot



Define Configuration Parameters for a SCARA Independent Robot

RSLogix 5000 software can be configured for control of robots with varying reach and payload capacities. As a result, it is very important to know the configuration parameter values for your robot including:

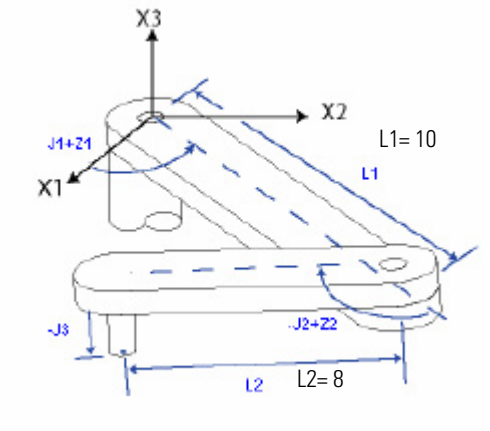
- link lengths
- base offsets
- end-effector offsets

The configuration information is available from the robot manufacturer.

IMPORTANT Be sure that the values for the link lengths, base offsets, and end-effector offsets are entered into the Configuration Parameters dialog using the same measurement units.

The following example illustrates the typical configuration parameters for a SCARA Independent robot.

Figure 1 - SCARA Independent

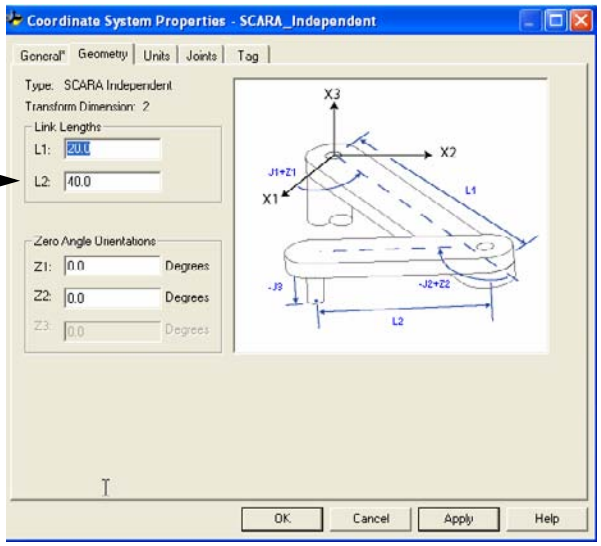


Link Lengths

Link lengths are the rigid mechanical bodies attached at joints.

Configuring Link Lengths for a SCARA Independent Robot

Enter the Link Length values.



For the robot shown in Figure 1 - SCARA Independent, the Link Length values are:

- L1 = 20
- L2 = 40

Base Offsets

Base offsets do not apply to a SCARA Independent robot configuration.

End-effector Offsets

End-effector offsets do not apply to a SCARA Independent robot configuration.

Configure Delta Robot Geometries

RSLogix 5000 software supports three types of geometries that are often called parallel manipulators.

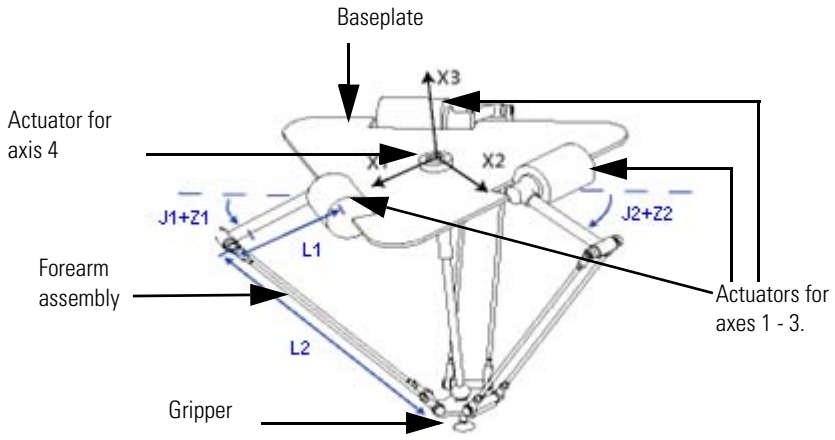
- Three-dimensional Delta
- Two-dimensional Delta
- SCARA Delta

In these geometries, the number of joints is greater than the degrees of freedom, and not all the joints are actuated (motor driven). These un-actuated joints are typically spherical joints.

Configure a Delta Three-dimensional Robot

This illustration shows a four axes Delta robot that moves in three-dimensional Cartesian (X1, X2, X3) space. This type of robot is often called a spider or umbrella robot.

Delta Three-dimensional Robot



The Delta robot in the above illustration is a three-degree of freedom robot with an optional fourth degree of freedom used to rotate a part at the tool tip. In RSLogix 5000 software, the first three-degrees of freedom are configured as three joint axes (J1, J2, J3) in the robots coordinate system. The three joint axes are either:

- directly programmed in joint space.
- automatically controlled by the embedded Kinematics software in RSLogix5000 software from instructions programmed in a virtual Cartesian coordinate system.

This robot contains a fixed top plate and a moving bottom plate. The fixed top plate is attached to the moving bottom plate by three link-arm assemblies. All three of the link-arm assemblies are identical in that they each have a single top link arm (L1) and a parallelogram two-bar link assembly (L2).

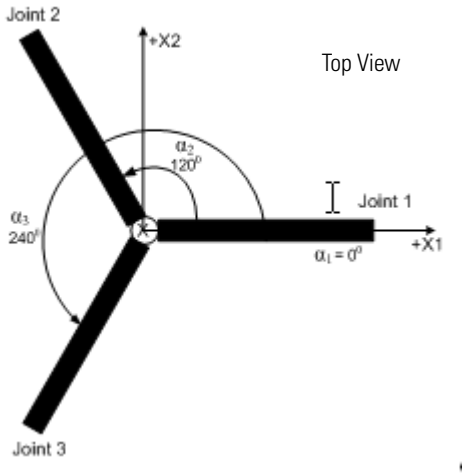
As each axis (J1, J2, J3) is rotated, the TCP of the gripper moves correspondingly in (X1, X2, X3) direction. The gripper remains vertical along the X3 axis while its position is translated to (X1, X2, X3) space by the mechanical action of the parallelograms in each of the three forearm assemblies. The mechanical connections of the parallelograms via spherical joints ensures that the top and bottom plates remain parallel to each other.

You program the TCP to an (X1, X2, X3) coordinate then, RSLogix 5000 software computes the commands necessary for each of the joints (J1, J2 ,J3) to move the gripper linearly from the current (X1, X2, X3) position to the programmed (X1, X2, X3) position, at the programmed vector dynamics.

When each top link (L1) moves downward, its corresponding joint axis (J1, J2, or J3) is assumed to be rotating in the positive direction. The three joint axes of the robot are configured as linear axes.

To rotate the gripper, configure a fourth axis as either a linear or rotary, independent axis.

Establish the Reference Frame for a Delta Three-dimensional Robot



The reference frame for the Delta geometries is located at the center of the top fixed plate. Joint 1, Joint 2, and Joint 3 are actuated joints. If you configure the Delta coordinate system in RSLogix 5000 with the joints homed at 0° in the horizontal position, then L1 of one of the link pairs will be aligned along the X1 positive axis as shown. Moving in the counter clockwise direction from Joint 1 to Joint 2, the X2 axis will be orthogonal to the X1 axis.

Based on the right hand rule, X3 positive will be the axis pointing up (out of the paper).

Calibrate a Delta Three-dimensional Robot

Use these steps to calibrate your robot.

1. Obtain the angle values from the robot manufacturer for J1, J2, and J3 at the calibration position. These values are used to establish the reference position.

2. Move all joints to the calibration position by either jogging the robot under programmed control, or manually moving the robot when the joint axes are in an open loop state.

3. Do **one** of these:
   a. Use a Motion Redefine Position instruction (MRP) to set the positions of the joint axes to the calibration values obtained in step 1.
   b. Set the configuration value for the joint axes home position to the calibration values obtained in step 1 and execute a Motion Axis Home instruction (MAH) for each joint axis.

4. Move each joint to an absolute position of 0.0. Verify that each joint position reads 0° and that the respective L1 is in a horizontal position. If L1 is not in a horizontal position, then refer to the alternate method for calibrating a Delta three-dimensional robot.

Alternate Method for Calibrating a Delta Three-dimensional Robot

Rotate each joint to a position so that the respective link is at a horizontal position, then perform **one** of the following:

a. Use a MRP instruction to set all the joint angles to 0° at this position.
b. Configure the values for the Zero Angle Offsets to be equal to the values of the joints when in a horizontal position.

Configure Zero Angle Orientations for Delta Three-dimensional Robot

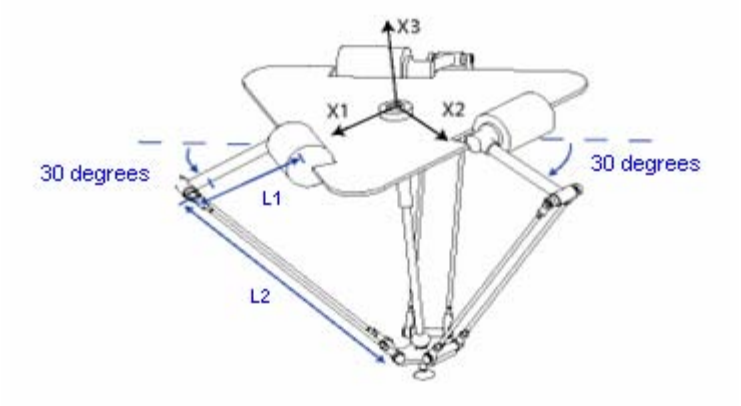
For Delta robot geometries, the internal transformation equations in the RSLogix 5000 software are written assuming that:

- joints are at 0° when link L1 is horizontal.
- as each top link (L1) moves downward, its corresponding joint axis (J1, J2, or J3) is rotating in the positive direction.

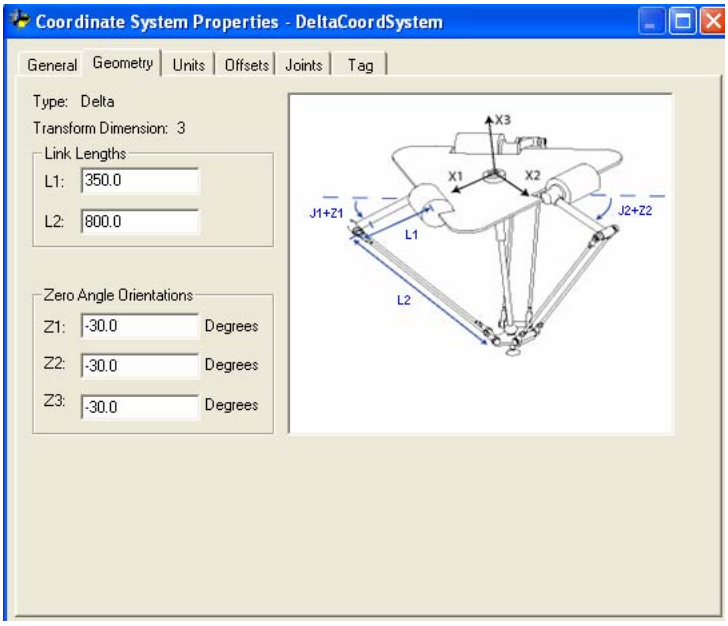
If you want the joint angular position when L1 is horizontal to be at any other value than 0°, then you must properly configure the Zero Angle Orientation values on the Geometry tab of the Target Coordinate System Properties dialog to align your joint angle positions with the internal equations.

For example, if your Delta robot is mounted so that the joints attached at the top plate are homed at 30° in the positive direction below horizontal (see Delta Robot with Joints Homed at 30° illustration) and you want the RSLogix 5000 software readout values to be zero in this position, then you must configure the Zero Angle Orientation values to -30° on the Geometry tab of the Target Coordinate System Properties dialog (see the Configuring Delta Robot Zero Angle Orientation illustration).

Delta Robot with Joints Homed at 30°



Configuring Delta Robot Zero Angle Orientation



Identify the Work Envelope for a Delta Three-dimensional Robot

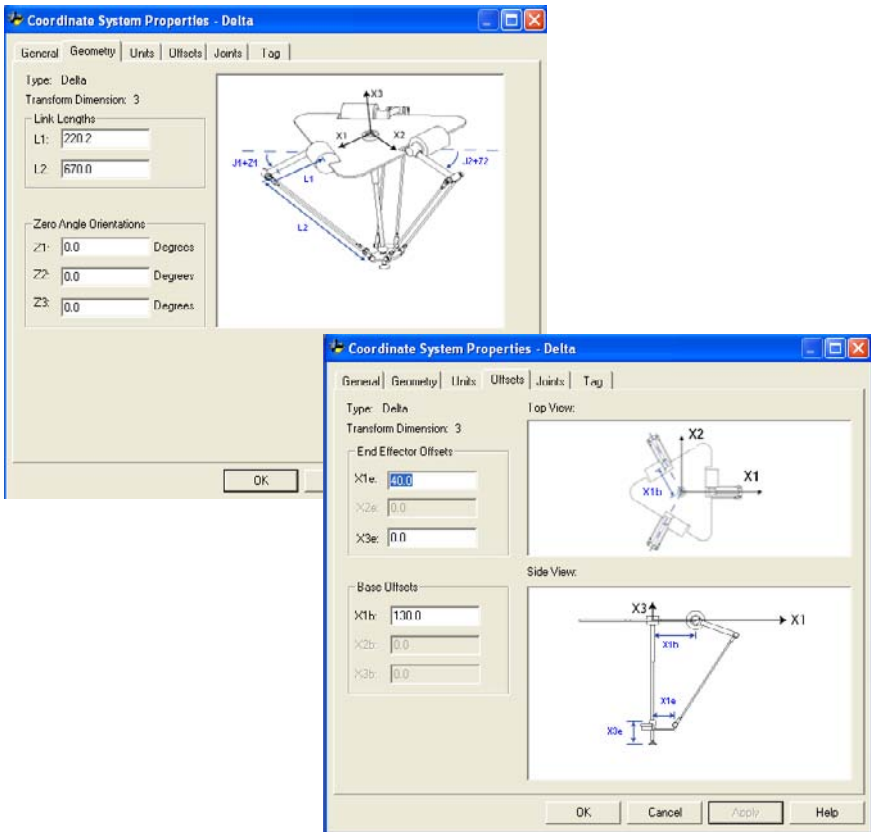
The work envelope is the three-dimensional region of space that defines the reaching boundaries for the robot arm. The typical work envelope for a Delta robot can be described as looking similar to plane in the upper region, with sides similar to a hexagonal prism and the lower portion similar to a sphere. For more detailed information regarding the work envelope of your Delta three-dimensional robot, refer to the documentation provided by the robot manufacturer.

We recommend that you program the robot within a rectangular solid defined inside the robots work zone. The rectangular solid can be defined by the positive and negative dimensions of the X1, X2, X3 virtual source axes. Be

sure that the robot position does not go outside the rectangular solid. You can check the position in the event task.

To avoid issues with singularity positions, RSLogix 5000 software internally calculates the joint limits for the Delta robot geometries. When an MCT instruction is invoked for the first time, the maximum positive and maximum negative joint limits are internally calculated based upon the link lengths and offset values entered on the Geometry and Offsets tabs of the Coordinate System Properties dialog.

Delta Three-dimensional Configuration Systems Properties Dialog - Geometry and Offsets Tabs



During each scan RSLogix5000 software evaluates the joint positions in the forward and inverse kinematics routines to be sure that they don't violate the computed maximum positive and maximum negative joint limits.

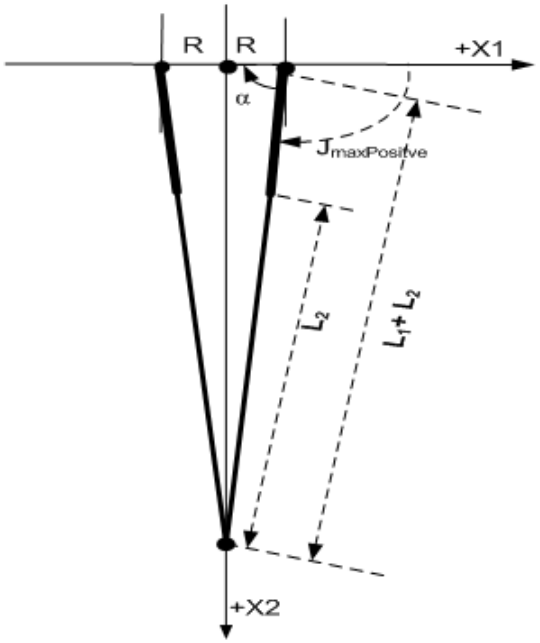
For More Information About	See Page
Maximum positive joint limits	114
Maximum negative joint limits	114

Homing or moving a joint axis to a position beyond a computed joint limit and then invoking a MCT instruction, results in an error 67 (Invalid Transform position). For more information regarding error codes refer to the Logix5000 Controllers Motion Instructions Reference Manual, publication 1756-RM007.

Maximum Positive Joint Limit Condition

The derivations for the maximum positive joint applies to the condition when L1 and L2 are collinear.

Maximum Positive Joint Limit Condition - L1 and L2 are Collinear



Maximum Positive Joint Limit Position

R = absolute value of (X1b - X1e)

$$\alpha = \cos^{-1}\left(\frac{R}{L1 + L2}\right)$$

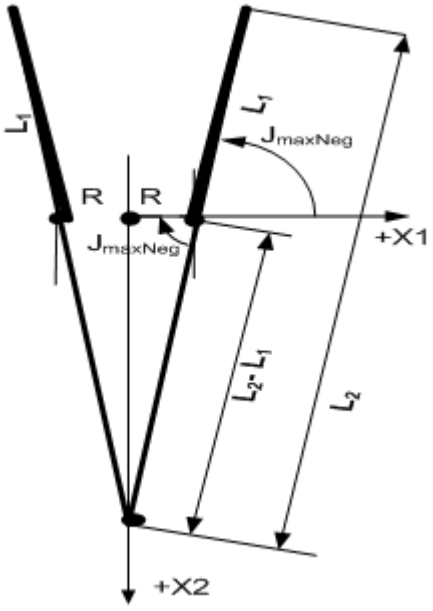
Jmax Positive = $180^\circ - \alpha$

Maximum Negative Joint Limit Condition

The derivations for the maximum negative joint limit applies to the condition when L1 and L2 are folded back on top of each other.

R is computed using the base and end-effector offsets values (X1b and X1e).

Maximum Negative Joint Limit Condition - L1 and L2 are Folded Back on Top of Each Other



Maximum Negative Joint Limit Condition

R = absolute value of (X1b - X1e)

$$JMaxNeg = -\cos^{-1}\left(\frac{R}{L2 - L1}\right)$$

Define Configuration Parameters for a Delta Three-dimensional Robot

RSLogix 5000 software can be configured for control of robots with varying reach and payload capacities. As a result, it is very important to know the configuration parameter values for your robot including:

- link lengths
- base offsets
- end-effector offsets

The configuration information is available from the robot manufacturer.

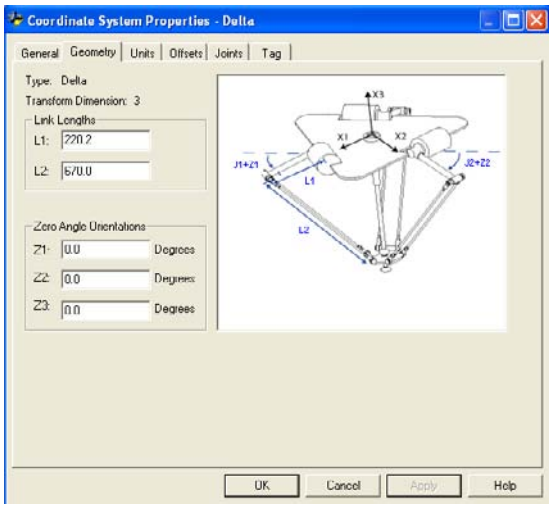
IMPORTANT Be sure that the values for the link lengths, base offsets, and end-effector offsets are entered into the Configuration Parameters dialog using the same measurement units.

Link lengths

Link lengths are the rigid mechanical bodies attached at the rotational joints. The three-dimensional Delta robot geometry has three link pairs each made up of L1 and L2. Each of the link pairs has the same dimensions.

- L1 - is the link attached to each actuated joint (J1, J2, and J3).
- L2 - is the parallel bar assembly attached to L1.

Three-dimensional Delta Robot - Link Lengths Configuration Screen



Base offsets

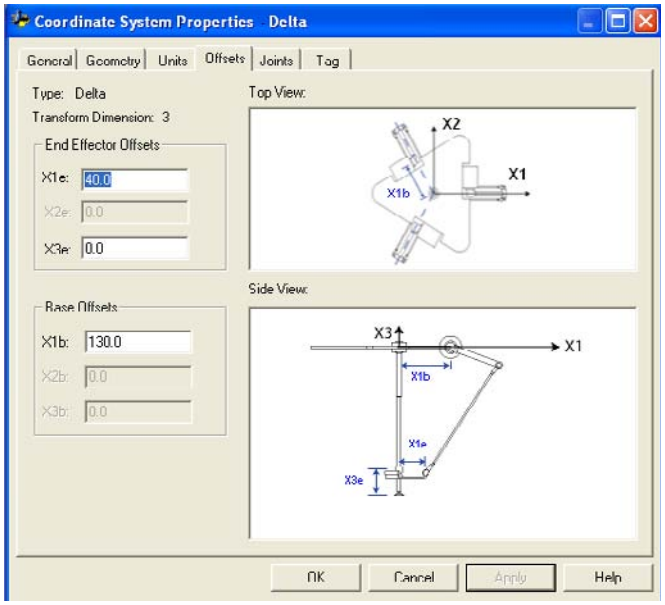
There is one base offset value (X1b) available for the three-dimensional Delta robot geometry. Enter the value equal to the distance from the origin of the robot coordinate system to one of the actuator joints.

End-effector offsets

The two end-effector offsets available for the three-dimensional Delta robot geometry are as follows. Offset values are always positive numbers.

- X1e is the distance from the center of the moving plate to the lower spherical joints of the parallel arms.
- X3e is the distance from the base plate to the TCP of the gripper.

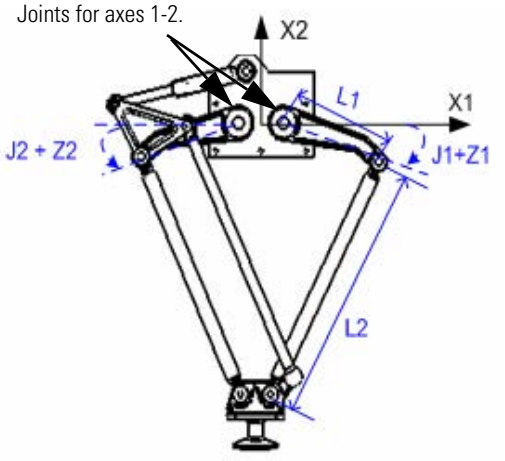
Configuring the Base Offset and End-effector Offsets for a Three-dimensional Delta Robot



Configure a Delta Two-dimensional Robot

This illustration shows a two-dimensional Delta robot that moves in two-dimensional Cartesian space.

Two-dimensional Delta Robot



This robot has two rotary joints that move the gripper in the (X1, X2) plane. Two, forearm assemblies attach a fixed top plate to a movable bottom plate. A gripper is attached to the movable bottom plate. The bottom plate is always orthogonal to the X2 axis and its position is translated in Cartesian space (X1, X2) by mechanical parallelograms in each forearm assembly. The two joints,

J1, and J2 are actuated joints. The joints between links L1 and L2 and between L2 and the base plate are unactuated joints.

Each joint is rotated independently to move the gripper to a programmed (X1, X2) position. As each joint axis (J1 and/or J2) is rotated, the TCP of the gripper moves correspondingly in the X1 and/or X2 direction. You can program the TCP to a (X1, X2) coordinate then RSLogix 5000 software uses internal vector dynamic calculations to compute the proper commands needed for each joint to move the gripper linearly from the current (X1, X2) position to the programmed (X1, X2) position.

The two joint axes (J1 and J2) of the robot are configured as linear axes.

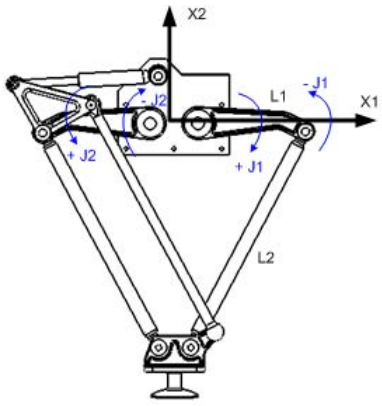
To rotate the gripper, configure a third axis as a linear or rotary, independent axis.

Establish the Reference Frame for a Delta Two-dimensional Robot

The reference frame for the two-dimensional Delta geometry is located at the center of the fixed top plate. When the angles of joints J1 and J2 are both at 0° , each of the two L1 links is along the X1 axis. One L1 link is pointing in the positive X1 direction, the other in the negative X1 direction.

When the right-hand link L1 moves downward, joint J1 is assumed to be rotating in the positive direction and when L1 moves upward, the J1 is assumed to be moving in the negative direction. When the left-hand link L1 moves downward, joint J2 is assumed to be rotating in the positive direction and when left-hand L1 moves upward the J2 is assumed to be moving in the negative direction.

Establishing the Two-dimensional Delta Robot Reference Frame



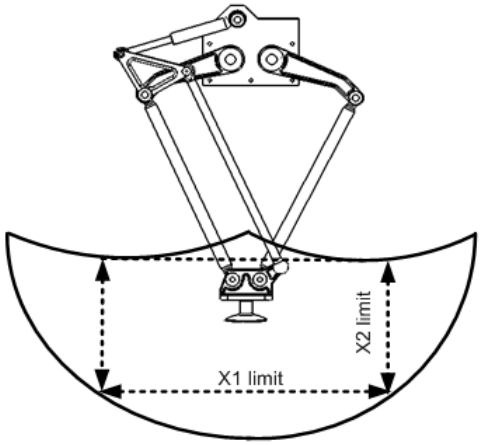
Calibrate a Delta Two-dimensional Robot

The method used to calibrate a Delta two-dimensional robot is the same as the method used for calibrating a Delta three-dimensional robot. The only difference is the number of axes used. For more information about calibration, refer to the section entitled Calibrate a Delta Three-dimensional Robot on page 110 of this manual.

Identify the Work Envelope for a Delta Two-dimensional Robot

The work envelope is the two-dimensional region of space that defines the reaching boundaries for the robot arm. The typical working envelope for a two-dimensional Delta robot is a boundary composed of circular arcs.

Work Envelope for Two-dimensional Delta Robot



We recommend that you define the program parameters for the two-dimensional Delta robot within a rectangle (dotted lines in the figure above) inside the robots work zone. The rectangle can be defined by the positive and negative dimensions of the X1, X2 virtual source axes. Be sure that the robot position does not go outside the rectangle. You can check the position in the event task.

To avoid problems with singularity positions, RSLogix 5000 software internally calculates the joint limits for the Delta robot geometries. When an MCT instruction is invoked for the first time, the maximum positive and maximum negative joint limits are internally calculated based upon the link lengths and offset values entered on the Geometry and Offsets tabs of the

Coordinate System Properties dialog Homing or moving a joint axis to a

For More Information About	See Page
Maximum positive joint limits	114
Maximum negative joint limits	114

position beyond a computed joint limit and then invoking a MCT instruction, results in an error 67 (Invalid Transform position). For more information regarding error codes refer to the Logix5000 Controllers Motion Instructions Reference Manual, publication 1756-RM007.

Define Configuration Parameters for a Delta Two-dimensional Robot

RSLogix 5000 software can be configured for control of robots with varying reach and payload capacities. As a result, it is very important to know the configuration parameter values for your robot including:

- link lengths
- base offsets
- end-effector offsets

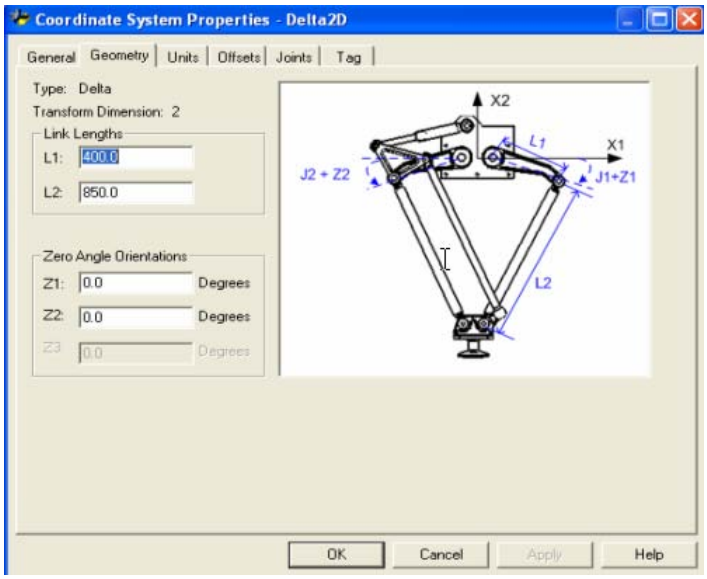
The configuration information is available from the robot manufacturer.

IMPORTANT Be sure that the values for the link lengths, base offsets, and end-effector offsets are entered into the Configuration Parameters dialog using the same measurement units.

Link lengths

Links are the rigid mechanical bodies attached at joints. The two- dimensional Delta geometry has two link pairs each with the same lengths. The link attached to each actuated joint (J1 and J2) is L1. The parallel bar assembly attached to link L1 is link L2.

Two-dimensional Delta Robot - Link Lengths Configuration Screen



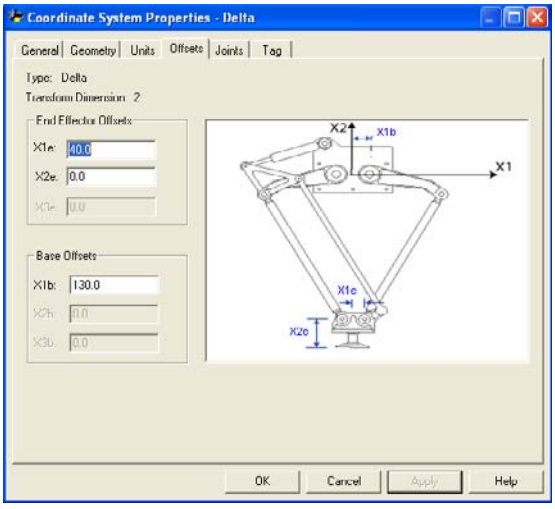
Base offsets

There is one base offset (X1b) available for the two-dimensional Delta robot geometry. Enter the value equal to the distance from the origin of the robot coordinate system to one of the actuator joints.

End-effector offsets

There are two end-effector offsets available for the two-dimensional Delta robot geometry. The value for X1e is the offset distance from the center of the lower plate to the lower spherical joints of the parallel arms. The distance from the lower plate to the TCP of the gripper is the value for X2e.

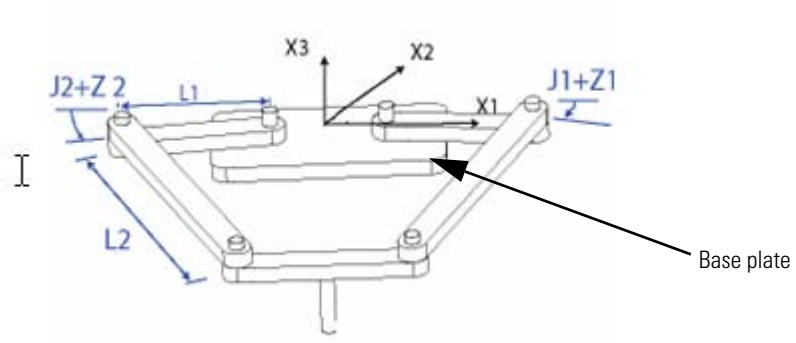
Delta Two-dimensional Robot - Base and End-effector Offsets



Configure a SCARA Delta Robot

The SCARA Delta robot geometry is similar to a two-dimensional Delta robot geometry except that the X1-X2 plane is tilted horizontally with the third linear axis in the vertical direction (X3).

SCARA Delta Robot



Establish the Reference Frame for a SCARA Delta Robot

The reference frame for the SCARA Delta robot is located at the center of the base plate.

When the angles of joints J1 and J2 are both at 0° , each of the two L1 links is along the X1 axis. One L1 link is pointing in the positive X1 direction, the other in the negative X1 direction.

When the right-hand link L1 moves in the clockwise direction (looking down on the robot), joint J1 is assumed to be rotating in the positive direction. When the right-hand link L1 moves counterclockwise, joint J1 is assumed to be moving in the negative direction.

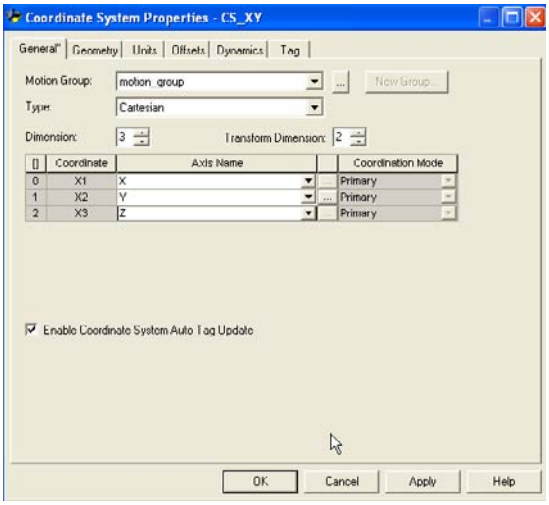
When left-hand link L1 moves in the clockwise direction, joint J2 is assumed to be moving in the negative direction.When the left-hand link L1 moves in the counterclockwise direction, joint J2 is assumed to be rotating in the positive direction.

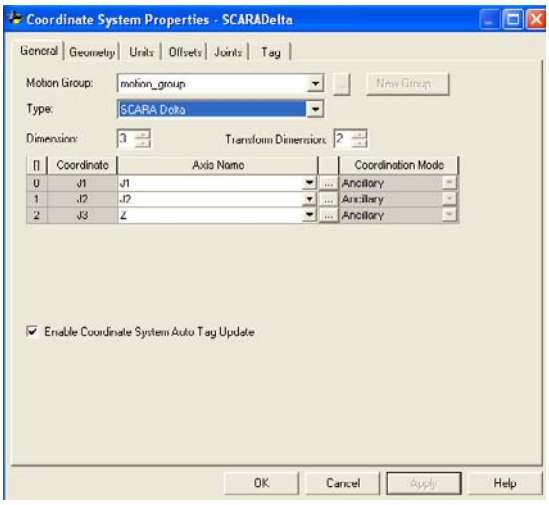
Based on the right hand rule, X3 positive will be orthogonal to the X1-X2 plane pointing up. The linear axis will always move in the X3 direction.

When configuring a SCARA Delta robot in RSLogix 5000 software keep the following in mind.

- Configure **both** the source and the target coordinate system with a transform dimension of two.
- The linear axis configured as a third axis must be the same for both the source and target coordinate systems.

Example of Source and Target Coordinate System Configuration for a SCARA Delta Robot





Calibrate a SCARA Delta Robot

The method used to calibrate a SCARA Delta robot is the same as the method used for calibrating a Delta three-dimensional robot. The only difference is the number of axes used. For more information about calibration, refer to the section entitled Calibrate a Delta Three-dimensional Robot on page 110 of this manual.

Identify the Work envelope for a SCARA Delta Robot

The work envelope for a SCARA Delta robot is similar to the two-dimensional Delta robot in the X1-X2 plane. The third linear axis extends the work region making it a solid region. The maximum positive and negative limits of the linear axis defines the height of the solid region.

We recommend that you program the SCARA Delta robot within a rectangular solid defined inside the robots work zone. The rectangular solid can be defined by the positive and negative dimensions of the X1, X2, X3 virtual source axes. Be sure that the robot position does not go outside the rectangular solid. You can check the position in the event task.

To avoid problems with singularity positions, RSLogix 5000 software internally calculates the joint limits for the Delta robot geometries. Homing or

For More Information About	See Page
Maximum positive joint limits	114
Maximum negative joint limits	114

moving a joint axis to a position beyond a computed joint limit and then invoking a MCT instruction, results in an error 67 (Invalid Transform position). For more information regarding error codes refer to the Logix5000 Controllers Motion Instructions Reference Manual, publication 1756-RM007.

Define Configuration Parameters for a SCARA Delta Robot

RSLogix 5000 software can be configured for control of robots with varying reach and payload capacities. As a result, it is very important to know the configuration parameter values for your robot including:

- link lengths
- base offset
- end-effector offset

The configuration information is available from the robot manufacturer.

IMPORTANT Be sure that the values for the link lengths, base offsets, and end-effector offsets are entered into the Configuration Parameters dialog using the same measurement units.

Link lengths

Links are the rigid mechanical bodies attached at joints. The SCARA Delta robot has two link pairs each with the same lengths. The link attached to each actuated joint (J1 and J2) is L1. The parallel bar assembly attached to link L1 is link L2.

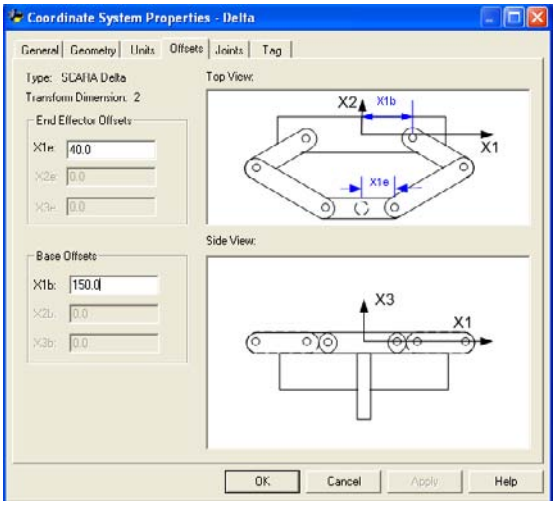
Base offset

There is one base offset (X1b) available for the SCARA Delta robot geometry. Enter the value equal to the distance from the origin of the robot coordinate system to one of the actuator joints. The base offset value is always a positive number.

End-effector offset

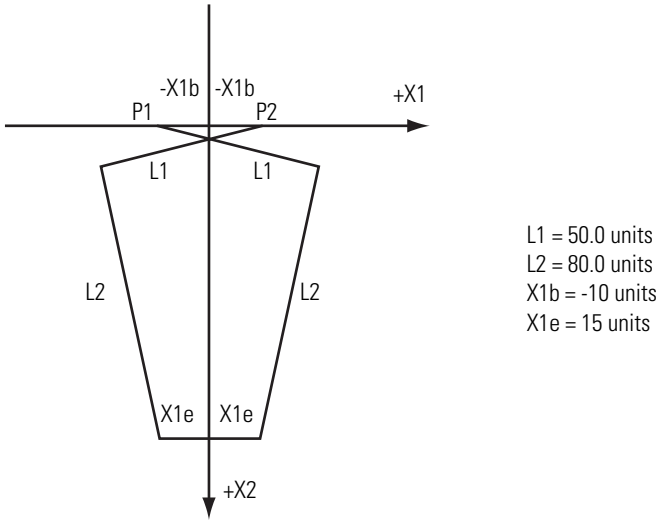
There is one end-effector offset (X1e) available for the SCARA Delta robot geometry. Enter the value for the distance from the center of the moving plate to one of the spherical joints of the parallel arms. The end-effector value is always a positive number.

SCARA Delta End-effector and Base offset



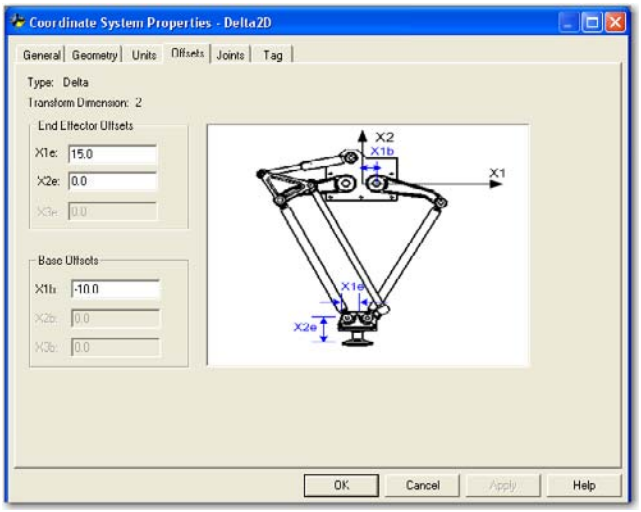
Configure a Delta Robot With a Negative X1b Offset

Beginning with RSLogix 5000, version 17, you can use negative offsets for the X1b base offset on both 2D and 3D delta geometries. For example, a mechanical 2D delta robot using a negative X1b offset has a mechanical configuration like the one shown below.



The base offset X1b is the value equal to the distance from the origin of the robot coordinate system to one of the actuator joints. In the previous figure, one of the actuator joints (P1), is on the negative side of X1. Therefore, the base offset X1b is measured to be a value of -10 units from the origin of the coordinate system (X1 - X2 intersection) to P1.

The RSLogix 5000 coordinate system configuration for the offset tab used with the example above is shown below.



This negative offset description also applies for Delta 3D and SCARA-Delta Configurations.

Arm Solutions

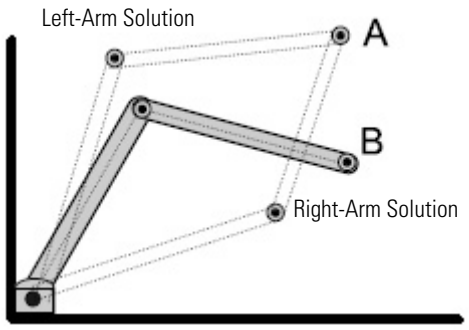
A Kinematic arm solution is the position of all joints on the robot that correspond to a Cartesian position. When the Cartesian position is inside the workspace of the robot then at least one solution will always exist. Many of the geometries have multiple joint solutions for a single Cartesian position.

- Two axis robots - two joint solutions typically exist for a Cartesian position.
- Three axis robots - four joint solutions typically exist for a Cartesian position.

Left-Arm and Right-Arm Solutions for Two-Axes Robots

A robot having an arm configuration has two Kinematics solutions when attempting to reach a given position (point A shown on the figure below). One solution satisfies the equations for a right-armed robot, the other solution satisfies the equations for a left-armed robot.

Right Arm and Left Arm Solutions



Solution Mirroring for Three-dimensional Robots

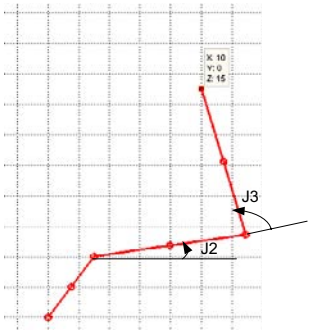
For a three-dimensional Articulated Independent robot there are four possible solutions for the same point.

- Left-Arm
- Right-Arm
- Left-Arm Mirror
- Right-Arm Mirror

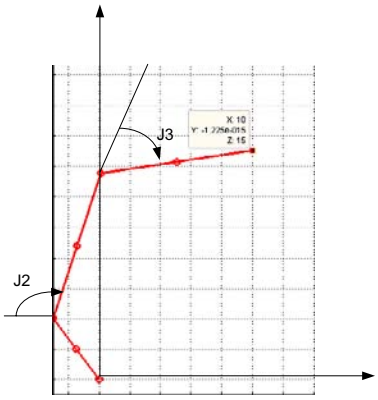
For example, consider the Cartesian point XYZ (10,0,15). The joint position corresponding to this point has four joint solutions. Two of the solutions are

the same as the solutions for the two-dimensional case. The other two solutions are mirror image solutions where J1 is rotated 180° .

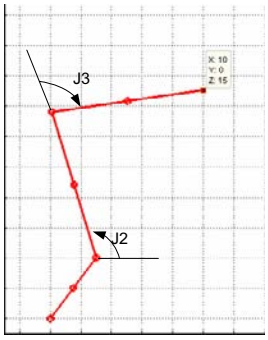
Right-Arm



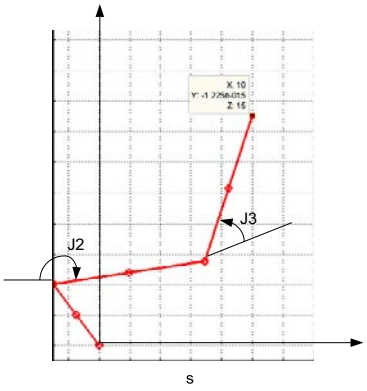
Right-Arm Mirror



Left-Arm



Left-Arm Mirror



Activating Kinematics

WARNING

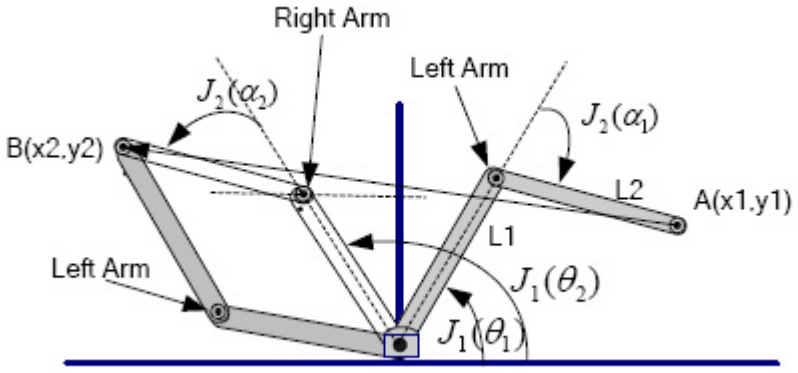
Be sure to choose an arm solution before activating the Kinematic function. Failure to do so can result in machine damage and/or serious injury or death to personnel.

Before activating Kinematics the robot should be in a left-arm or right-arm solution. The robot stays in the same configuration in which it was activated as it is moved in Cartesian or source coordinate mode. If activated in a fully-extended-arm mode (this is neither a left-arm nor a right-arm solution), the system chooses a left-arm solution.

Change the Robot Arm Solution

You can switch the robot from a left-arm solution to a right-arm solution or vice versa. This is done automatically when a joint move is programmed forcing a left/right change to occur. After the change is performed the robot stays in the new arm solution when Cartesian moves are made. The robot arm solution changes again (if required) when another joint move is made.

Example: Suppose, you want to move the robot from position A (x1,y1) to position B (X_2,Y_2) (refer to figure below). At position A the system is in a left arm solution. Programming a Cartesian move from A (X_1,Y_1) to B (X_2,Y_2) means that the system moves along the straight line (refer to illustration) from A to B while maintaining a left arm solution. If you want to be at position B in a right-arm solution, you must make a joint move in J1 from θ_1 to θ_2 and a joint move in J2 from α_1 to α_2.



Plan for Singularity

A singularity occurs when an infinite number of joint positions (mathematical solutions) exist for a given Cartesian position. The Cartesian position of a singularity is dependent on the type of the robot geometry and the size of the link lengths for the robot. Not all robot geometries have singularity positions.

For example, singularities for an Articulated Independent robot occur when:

- the robot manipulator folds its arm back onto itself and the Cartesian position is at the origin.
- the robot is fully stretched at or very near the boundary of its workspace.

An error condition is generated when a singularity position is reached.

WARNING

Avoid programming your robot towards a singularity position when programming in Cartesian mode. The velocity of the robot increases very rapidly as it approaches a singularity position and can result in injury or death to personnel.

Encounter a No-solution Position

WARNING

Avoid programming your robot towards a no solution position when programming in Cartesian mode. The velocity of the robot increases very rapidly as it approaches this position and can result in injury or death to personnel.

When a robot is programmed to move beyond its work envelope, there is no mathematical joint position for the programmed Cartesian position. The system forces an error condition.

For example, if an Articulated Independent robot has two 10-inch arms, the maximum reach is 20 inches. Programming to a Cartesian position beyond 20 inches produces a condition where no mathematical joint position exists.

Error Conditions

Kinematics error conditions are detected:

- upon activation of a transformation by executing a MCT instruction.
- in some movement conditions.

  Errors can occur for certain movement conditions for either the source or target coordinate system after a transformation has been established. These type of errors are reported in the MCT instruction error codes. Singularity and other movement error conditions are also reported in the MCT error codes.

- computing an invalid position via an MCTP instruction.

For a list and description of error codes, refer to the LOGIX5000 Controllers Motion Instructions, publication 1756-RM007.

Monitor Status Bits for Kinematics

You can monitor the status of the Kinematics functions using RSLogix 5000 software status bits.

To see if	Check the following tag	And this bit	For
A coordinate system is the source of an active transform	Coordinate system	TransformSourceStatus	On
A coordinate system is the target of an active transform	Coordinate system	TransformTargetStatus	On
An axis is part of an active transform	Axis	TransformStateStatus	On
An axis is moving because of a transform	Axis	ControlledByTransformStatus	On

Notes:

Interpret Module Lights (LEDs)

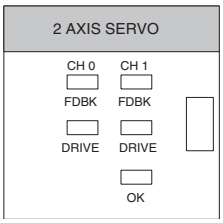
Introduction

Use this chapter to interpret the lights on the front of your module.

For This Module	See Page
1756-M02AE Module	133
1756-M02AS Module	135
1756-HYD02 Module	138
SERCOS interface Module	141

1756-M02AE Module

OK Light



State	Description	Recommended Action
Off	The module is not operating.	· Apply chassis power. · Verify the module is completely inserted into the chassis and backplane.
Flashing green	The module has passed internal diagnostics, but it is not communicating axis data over the backplane.	· None, if you have not configured the module. · If you have configured the module, check the slot number in the 1756-M02AE Properties dialog box.
Steady green	· Axis data is being exchanged with the module. · The module is in the normal operating state.	None. The module is ready for action.
Flashing red	· A major recoverable failure has occurred. · A communication fault, timer fault, or NVS update is in progress. · The OK contact has opened.	· Check the servo fault word for the source of the error. · Clear the fault condition using the motion instructions. · Resume normal operation. · If the flashing persists, reconfigure the module.
Solid red	· A potential non-recoverable fault has occurred. · The OK contact has opened.	· Reboot the module. · If the solid red persists, replace the module.

FDBK Light

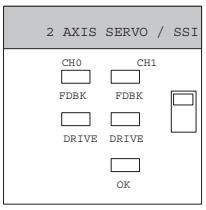
State	Description	Recommended Action
Off	The axis is not used.	· None, if you are not using this axis. · If you are using this axis, make sure you configured the module and associated an axis tag with the module.
Flashing green	The axis is in the normal servo loop inactive state.	None. You can change the servo axis state by executing motion instructions.
Steady green	The axis is in the normal servo loop active state.	None. You can change the servo axis state by executing motion instructions.
Flashing red	The axis servo loop error tolerance has been exceeded.	· Correct the source of the problem. · Clear the servo fault using a fault reset instruction. · Resume normal operation.
Solid red	An axis encoder feedback fault has occurred.	· Correct the source of the problem by checking the encoder and power connections. · Clear the servo fault using the MAFR instruction. · Resume normal operation.

DRIVE Light

State	Description	Recommended Action
Off	· The axis is not used. · The axis is a position-only axis type.	· None, if you are not using the axis or have configured it as a position-only axis. · Otherwise, make sure you have configured the module, associated an axis tag with the module, and configured the axis as a servo axis.
Flashing green	The axis drive is in the normal disabled state.	None. You can change the servo axis state by executing a motion instruction.
Steady green	The axis drive is in the normal enabled state.	None. You can change the servo axis state by executing a motion instruction.
Flashing red	The axis drive output is in the Shutdown state.	· Check for faults that may have generated this state. · Execute the shutdown reset motion instruction. · Resume normal operation.

Solid red	The axis drive is faulted.	· Check the drive status.
		· Clear the drive fault condition at the drive.
		· Execute a fault reset motion instruction.
		· Resume normal operation.
		· Check the configuration for the Drive Fault.
		  · If configured to be normally open and there is no voltage, this is the normal condition.
		  · If configured to be normally closed and there is 24V applied, this is the normal condition.

1756-M02AS Module OK Light



State	Description	Recommended Action
Off	The module is not operating.	· Apply chassis power. · Verify the module is completely inserted in chassis and backplane.
Flashing green	The module has passed internal diagnostics, but it is not communicating axis data over the backplane.	None, if you have not configured the module. If you have configured the module, check the slot number in the 1756-M02AS Properties dialog box.
Steady green	One of the following: · Module is exchanging axis data. · The module is in the normal operating state.	None
Flashing red	One of the following: · A major recoverable failure has occurred. · A communication fault, timer fault, or non-volatile memory storage (NVS) update is in progress. · The OK contact has opened.	If an NVS update is in progress, complete the NVS update. If an NVS update is not in progress: · Check the Servo Fault word for the source of the error. · Clear the servo fault condition via Motion Axis Fault Reset instruction. · Resume normal operation. · If the flashing persists, reconfigure the module.
Steady red	One of the following: · A potential non- recoverable fault has occurred. · The OK contact has opened.	Reboot the module. If the solid red persists, replace the module.

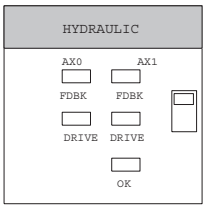
FDBK Light

State	Description	Recommended Action
Off	The axis is not used.	None, if you are not using this axis. If you are using this axis, make sure the module is configured and an axis tag has been associated with the module.
Flashing green	The axis is in the normal servo loop inactive state.	None. The servo axis state can be changed by executing motion instructions.
Steady green	The axis is in the normal servo loop active state.	None. The servo axis state can be changed by executing motion instructions.
Flashing red	The axis servo loop error tolerance has been exceeded.	· Correct the source of the problem. · Clear the servo fault condition using the Motion Axis Fault Reset instruction. · Resume normal operation.
Steady red	An axis SSI feedback fault has occurred.	· Correct the source of the problem by checking the SSI device and power connections. · Clear the servo fault condition using the Motion Axis Fault Reset instruction. · Resume normal operation.

DRIVE Light

State	Description	Recommended Action
Off	One of the following: · The axis is not used. · The axis is a position- only axis type.	None, if the axis is not used or is a position- only type. Otherwise, make sure the module is configured, an axis tag has been associated with the module, and the axis type is servo.
Flashing green	The axis drive is in the normal disabled state.	None. The servo axis state can be changed by executing motion instructions.
Steady green	The axis drive is in the normal enabled state.	None. The servo axis state can be changed by executing motion instructions.
Flashing red	The axis drive output is in the shutdown state.	· Check for faults that may have generated this state. · Execute the Motion Axis Shutdown Reset instruction. · Resume normal operation.
Steady red	The axis drive is faulted.	· Check the drive status. · Clear the Drive Fault condition at the drive. · Clear the servo fault condition using the Motion Axis Fault Reset instruction. · Resume normal operation. · Check the configuration for the Drive Fault. · If configured to be normally open and there is no voltage, this is the normal condition. · If configured to be normally closed and 24V dc is applied, this is the normal condition.

1756-HYD02 Module OK Light



State	Description	Recommended Action
Off	The module is not operating.	· Apply chassis power. · Verify the module is completely inserted in chassis and backplane.
Flashing green	The module has passed internal diagnostics, but it is not communicating axis data over the backplane.	None, if you have not configured the module. If you have configured the module, check the slot number in the 1756-HYD02 Properties dialog box.
Steady green	One of the following: · Module is exchanging axis data. · The module is in the normal operating state.	None
Flashing red	One of the following: · A major recoverable failure has occurred. · A communication fault, timer fault, or non-volatile memory storage (NVS) update is in progress. · The OK contact has opened.	If an NVS update is in progress, complete the NVS update. If an NVS update is not in progress: · Check the Servo Fault word for the source of the error. · Clear the servo fault condition via Motion Axis Fault Reset instruction. · Resume normal operation. · If the flashing persists, reconfigure the module.
Steady red	One of the following: · A potential non- recoverable fault has occurred. · The OK contact has opened.	Reboot the module. If the solid red persists, replace the module.

FDBK Light

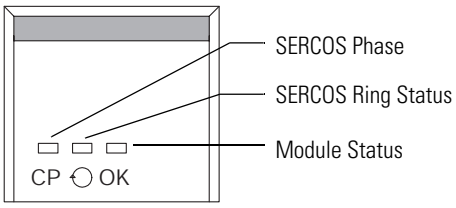
State	Description	Recommended Action
Off	The axis is not used.	None, if you are not using this axis. If you are using this axis, make sure the module is configured and an axis tag has been associated with the module.
Flashing green	The axis is in the normal servo loop inactive state.	None. The servo axis state can be changed by executing motion instructions.
Steady green	The axis is in the normal servo loop active state.	None. The servo axis state can be changed by executing motion instructions.
Flashing red	The axis servo loop error tolerance has been exceeded.	· Correct the source of the problem. · Clear the servo fault condition using the Motion Axis Fault Reset instruction. · Resume normal operation.
Steady red	An axis LDT feedback fault has occurred.y	· Correct the source of the problem by checking the LDT and power connections. · Clear the servo fault condition using the Motion Axis Fault Reset instruction. · Resume normal operation.

DRIVE Light

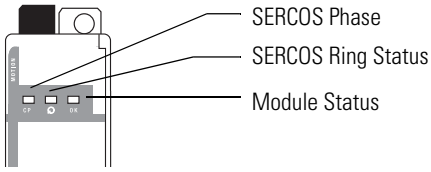
State	Description	Recommended Action
Off	One of the following: · The axis is not used. · The axis is a position- only axis type.	None, if the axis is not used or is a position- only type. Otherwise, make sure the module is configured, an axis tag has been associated with the module, and the axis type is servo.
Flashing green	The axis drive is in the normal disabled state.	None. The servo axis state can be changed by executing motion instructions.
Steady green	The axis drive is in the normal enabled state.	None. The servo axis state can be changed by executing motion instructions.
Flashing red	The axis drive output is in the shutdown state.	· Check for faults that may have generated this state. · Execute the Shutdown Reset motion instruction. · Resume normal operation.
Steady red	The axis drive is faulted.	· Check the drive status. · Clear the Drive Fault condition at the drive. · Clear the servo fault condition using the Motion Axis Fault Reset instruction. · Resume normal operation. · Check the configuration for the Drive Fault. · If configured to be normally open and there is no voltage, this is the normal condition. · If configured to be normally closed and 24V dc is applied, this is the normal condition.

SERCOS interface Module

1756-M03SE, 1756-M08SE, 1756-M16SE



1768-M04SE



If the lights on the module look like this			Then do this
CP	**Ring**	**OK**	
Off	Off	Off	· Make sure the module is all the way in the chassis or connected and locked to the other modules. · Is this a 1768-M04SE module? · No — Check the power supply for power. · Yes — Check the power supply and controller for power.
Off	Off	Flashing Red	Wait! Someone is updating the firmware of the module.
Flashing Orange	Off	Flashing Green	· Look for cables that are broken, unplugged, or in the wrong port. · Check the drives for faults.
Solid Orange	Flashing Red	Flashing Green	· Make sure each drive has its own address. · Make sure that all of the drives have the same baud rate. · Set the Data Rate of the SERCOS interface module to Auto-Detect. · Check the Cycle Time of the SERCOS interface module. See Specifications.
Flashing Red and Green	Flashing Green	Flashing Green	Did you configure the module? · NO — Use RSLogix 5000 software to configure the module. · YES — Check the configuration of the module and drives in RSLogix 5000 software.
Flashing Green	Flashing Green	Flashing Green	Check the configuration of the axes in RSLogix 5000 software.
Solid Green	Solid Green	Flashing Green	· Check the configuration of the drives in RSLogix 5000 software. · Check the motion group, drives, and axes for faults.
Solid Green	Solid Green	Solid Green	None — the axes are ready.
Solid Green	Solid Green	Flashing Red	Check the motion group and axes for faults.
Solid Red	Solid Red	Solid Red	1. Cycle power to the module. 2. If the lights keep turning solid red, contact your distributor, Rockwell Automation representative, or Rockwell Automation support.

Notes:

Troubleshoot Axis Motion

Introduction

This chapter helps you troubleshoot some situations that could happen while you are running an axis.

Situation	See page
Why does my axis accelerate when I stop it?	143
Why does my axis overshoot its target speed?	145
Why is there a delay when I stop and then restart a jog?	148
Why does my axis reverse direction when I stop and start it?	150

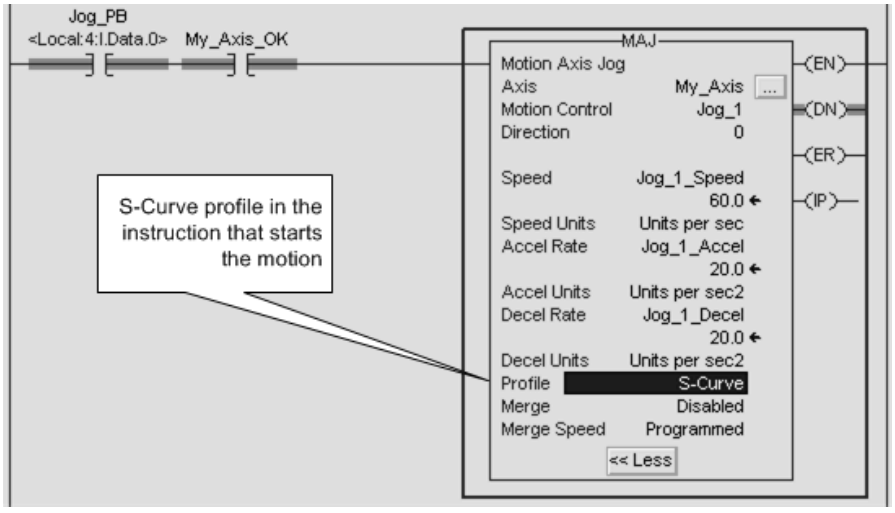
Why does my axis accelerate when I stop it?

While an axis is accelerating, you try to stop it. The axis keeps accelerating for a short time before it starts to decelerate.

Example

You start a Motion Axis Jog (MAJ) instruction. Before the axis gets to its target speed, you start a Motion Axis Stop (MAS) instruction. The axis continues to speed up and then eventually slows to a stop.

Look for



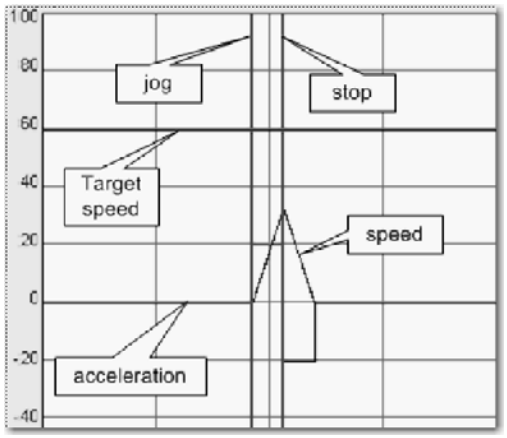
Cause When you use an S-curve profile, jerk determines the acceleration and deceleration time of the axis.

- An S-curve profile has to get acceleration to 0 before the axis can slow down.
- The time it takes depends on the acceleration and speed.
- In the meantime, the axis continues to speed up.

The following trends show how the axis stops with a trapezoidal profile and an S-curve profile.

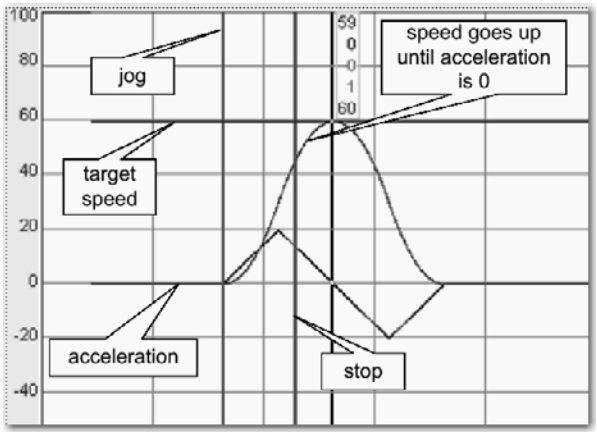
Stop while accelerating

Trapezoidal



The axis slows down as soon as you start the stopping instruction.

S-curve



The axis continues to speed up until the S-curve profile brings the acceleration rate to 0.

Corrective action If you want the axis to slow down right away, use a trapezoidal profile.

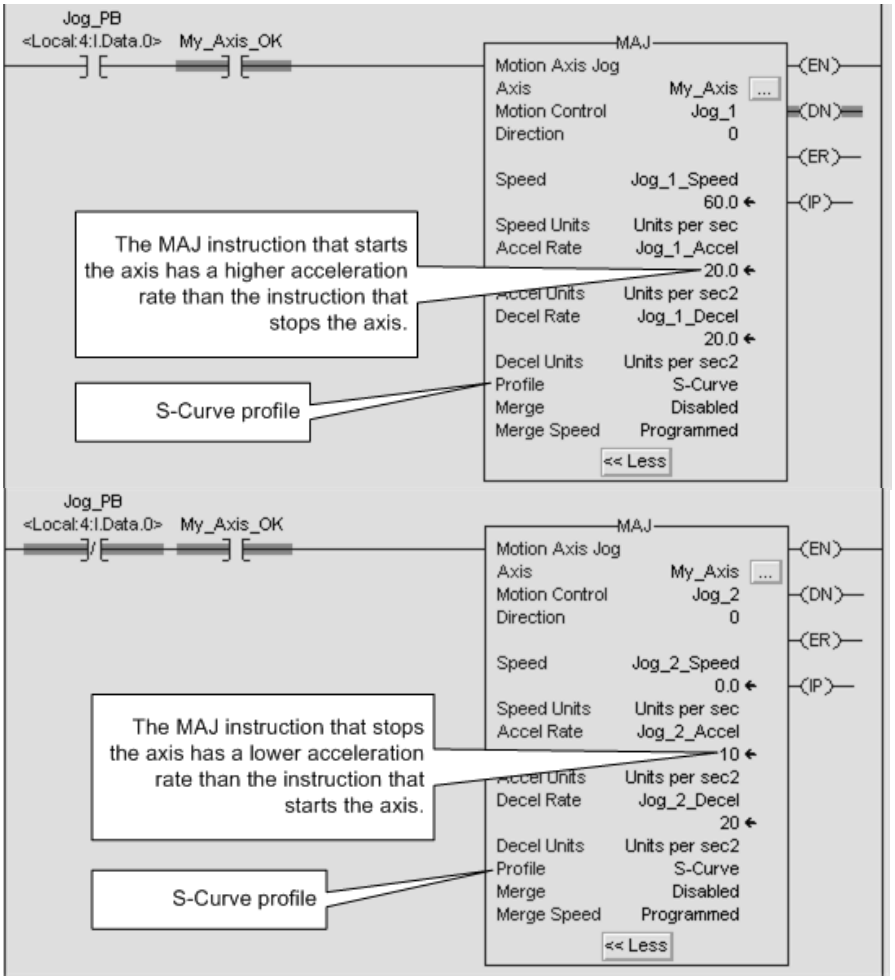
Why does my axis overshoot its target speed?

While an axis is accelerating, you try to stop the axis or change its speed. The axis keeps accelerating and goes past its initial target speed. Eventually it starts to decelerate.

Example

You start a Motion Axis Jog (MAJ) instruction. Before the axis gets to its target speed, you try to stop it with another MAJ instruction. The speed of the second instruction is set to 0. The axis continues to speed up and overshoots its initial target speed. Eventually it slows to a stop.

Look for



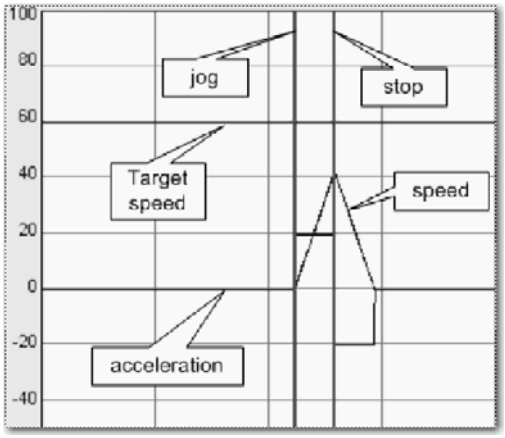
Cause When you use an S-curve profile, jerk determines the acceleration and deceleration time of the axis.

- An S-curve profile has to get acceleration to 0 before the axis can slow down.
- If you reduce the acceleration, it takes longer to get acceleration to 0.
- In the meantime, the axis continues past its initial target speed.

The following trends show how the axis stops with a trapezoidal profile and an S-curve profile.

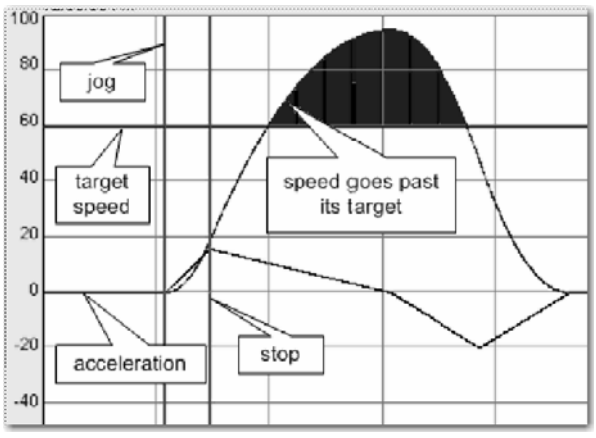
Stop while accelerating and reduce the acceleration rate

Trapezoidal



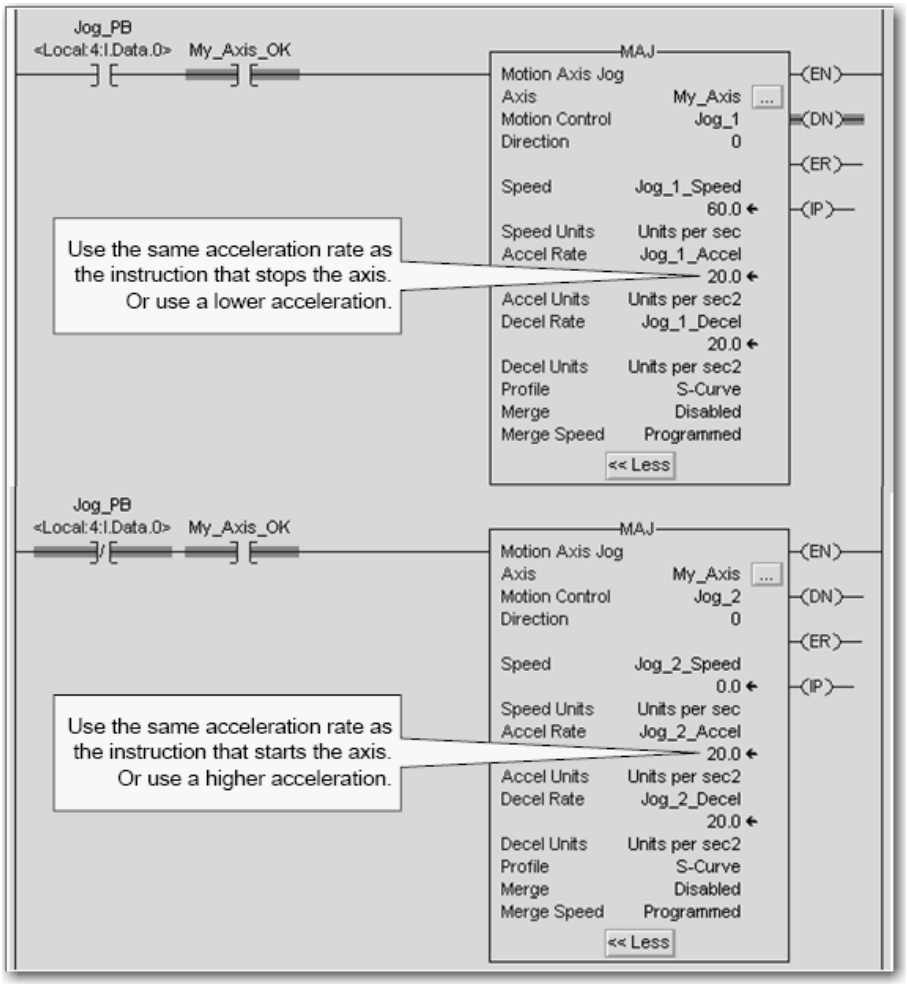
The axis slows down as soon as you start the stopping instruction. The lower acceleration doesn't change the response of the axis.

S-curve



The stopping instruction reduces the acceleration of the axis. It now takes longer to bring the acceleration rate to 0. The axis continues past its target speed until acceleration equals 0.

Corrective action Use a Motion Axis Stop (MAS) instruction to stop the axis or set up your instructions like this.



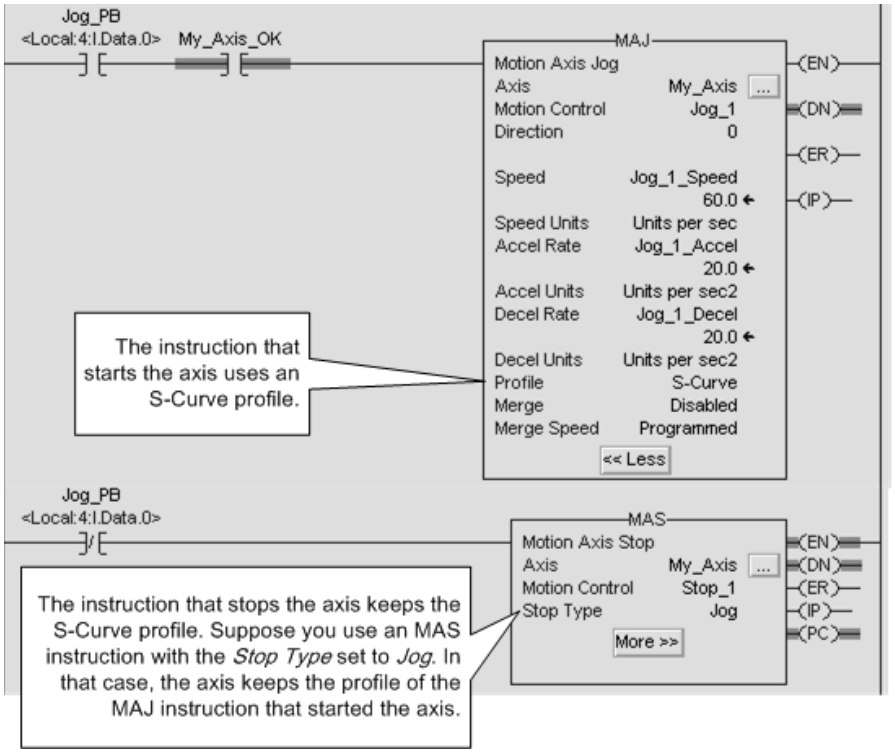
Why is there a delay when I stop and then restart a jog?

While an axis is jogging at its target speed, you stop the axis. Before the axis stops completely, you restart the jog. The axis continues to slow down before it speeds up.

Example

You use a Motion Axis Stop (MAS) instruction to stop a jog. While the axis is slowing down, you use a Motion Axis Jog (MAJ) instruction to start the axis again. The axis doesn't respond right away. It continues to slow down. Eventually it speeds back up to the target speed.

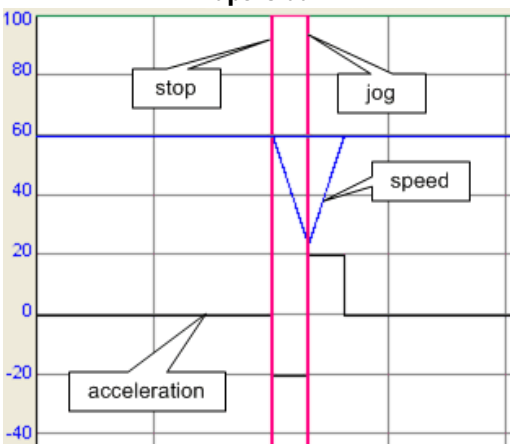
Look for



Cause When you use an S-curve profile, jerk determines the acceleration and deceleration time of the axis. An S-curve profile has to get acceleration to 0 before the axis can speed up again. The following trends show how the axis stops and starts with a trapezoidal profile and an S-curve profile.

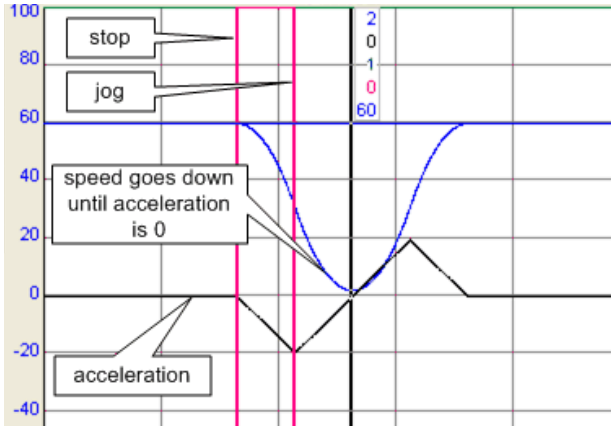
Start while decelerating

Trapezoidal



The axis speeds back up as soon as you start the jog again.

S-curve



The axis continues to slow down until the S-curve profile brings the acceleration rate to 0.

Corrective action If you want the axis to accelerate right away, use a trapezoidal profile.

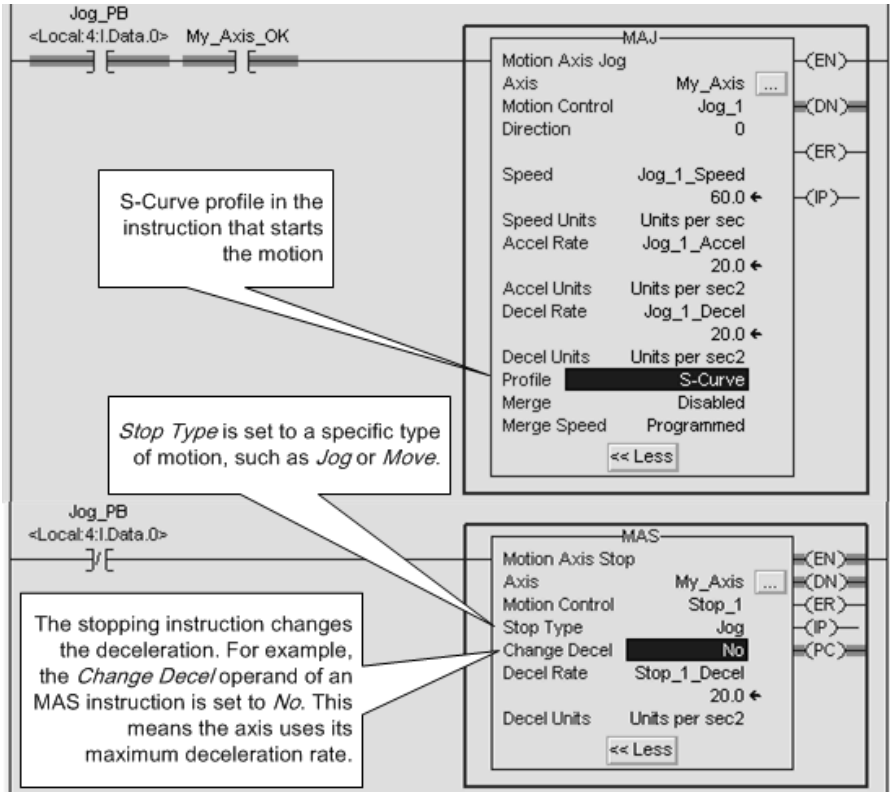
Why does my axis reverse direction when I stop and start it?

While an axis is jogging at its target speed, you stop the axis. Before the axis stops completely, you restart the jog. The axis continues to slow down and then reverse direction. Eventually the axis changes direction again and moves in the programmed direction.

Example

You use a Motion Axis Stop (MAS) instruction to stop a jog. While the axis is slowing down, you use a Motion Axis Jog (MAJ) instruction to start the axis again. The axis continues to slow down and then moves in the opposite direction. Eventually goes back to its programmed direction.

Look for



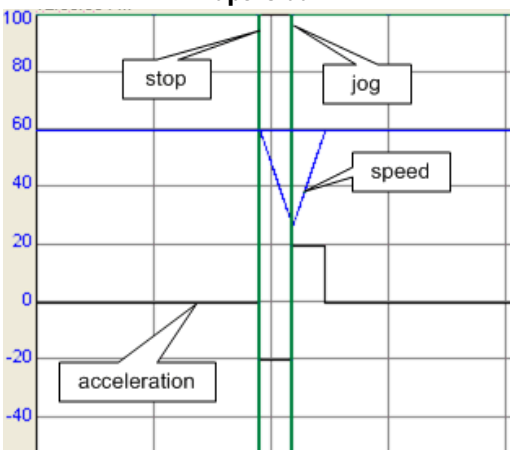
Cause When you use an S-curve profile, jerk determines the acceleration and deceleration time of the axis.

- An S-curve profile has to get acceleration to 0 before the axis can speed up again.
- If you reduce the acceleration, it takes longer to get acceleration to 0.
- In the meantime, the axis continues past 0 speed and moves in the opposite direction.

The following trends show how the axis stops and starts with a trapezoidal profile and an S-curve profile.

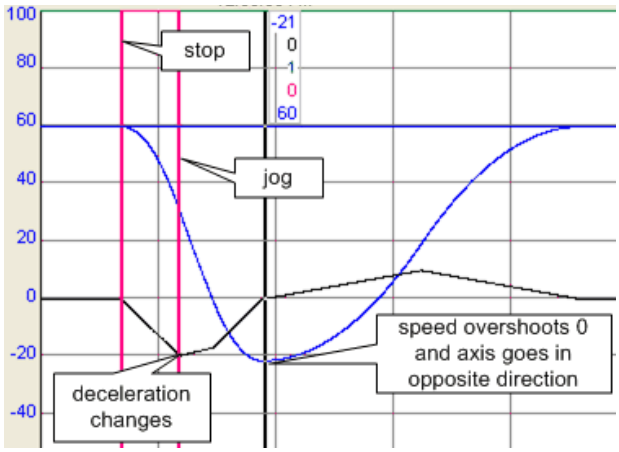
Start while decelerating and reduce the deceleration rate

Trapezoidal



The axis speeds back up as soon as you start the jog again. The lower deceleration doesn't change the response of the axis.

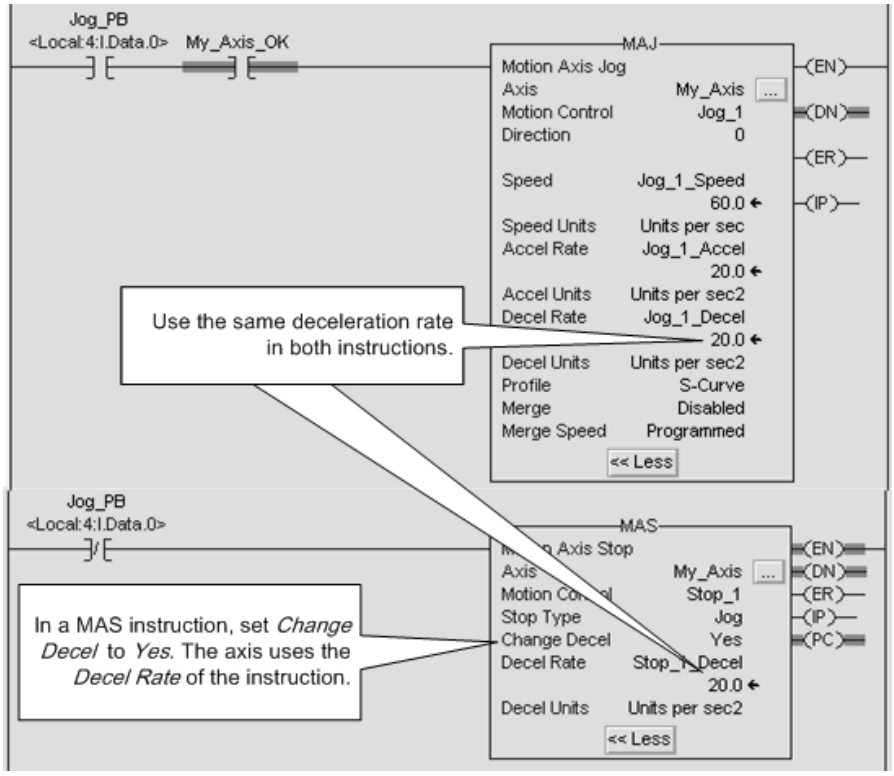
S-curve



The jog instruction reduces the deceleration of the axis. It now takes longer to bring the acceleration rate to 0. The speed overshoots 0 and the axis moves in the opposite direction.

Corrective action

Use the same deceleration rate in the instruction that starts the axis and the instruction that stops the axis.



Configure Homing

Introduction

Homing puts your equipment at a specific starting point for operation. This starting point is called the home position. Typically, you home your equipment when you reset it for operation.

Guidelines for Homing

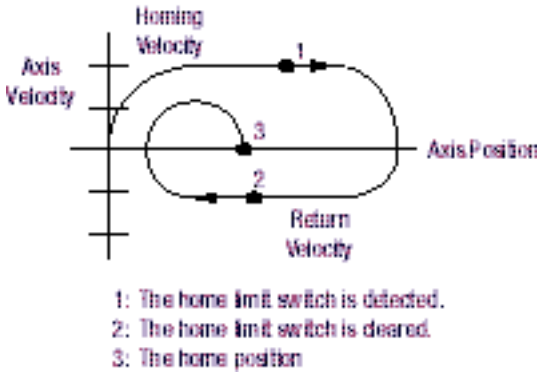
Guideline	Details
1.To move an axis to the home position, use Active homing.	Active homing turns on the servo loop and moves the axis to the home position. Active homing also: · Stops any other motion. · Uses a trapezoidal profile.
2.For a Feedback-only device, use Passive homing.	Passive homing doesn't move the axis. · Use passive homing to calibrate a Feedback-only axis to its marker. · If you use passive homing on a servo axis, turn on the servo loop and use a move instruction to move the axis.
3.If you have an absolute feedback device, consider Absolute homing.	If the motion axis hardware supports an absolute feedback device, Absolute Homing Mode may be used. The only valid Home Sequence for an absolute Homing Mode is Immediate. In this case, the absolute homing process establishes the true absolute position of the axis by applying the configured Home Position to the reported position of the absolute feedback device. Prior to execution of the absolute homing process via the MAH instruction, the axis must be in the Axis Ready state with the servo loop disabled.
4.For single-turn equipment, consider homing to a marker.	The marker homing sequence is useful for single-turn rotary and linear encoder applications because these applications have only one encoder marker for full axis travel.
5.For multi-turn equipment, home to a switch or switch and marker.	These homing sequences use a home limit switch to define the home position. · You need a home limit switch if the axis moves more than one revolution when it runs. Otherwise the controller can't tell which marker pulse to use. · For the most precise homing, use both the switch and marker.

6.If your equipment can't back up, use unidirectional homing.	With unidirectional homing, the axis doesn't reverse direction to move to the Home Position. For greater accuracy, consider using an offset.
	· Use a Home Offset that is in the same direction as the Home Direction.
	· Use a Home Offset that is greater than the deceleration distance.
	· If the Home Offset is less than the deceleration distance:
	  · The axis simply slows to a stop. The axis doesn't reverse direction to move to the Home Position. In this case, the MAH instruction doesn't set the PC bit.
	  · On a rotary axis, the controller adds 1 or more revolutions to the move distance. This makes sure that the move to the Home Position is unidirectional.

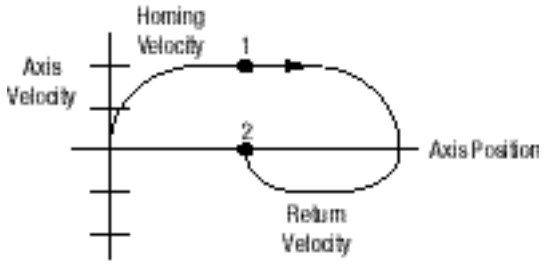
Guideline	Details
7.Choose a starting direction for the homing sequence.	Which direction do you want to start the homing sequence in? · Positive direction — choose a Forward direction. · Negative direction — choose a Negative direction.

Examples

Active Homing

Sequence	Description
Active immediate home	This sequence sets the axis position to the Home Position without moving the axis. If feedback isn't enabled, this sequence enables feedback.
Active home to switch in forward bidirectional	The switch homing sequence is useful for multi-turn rotary and linear applications.  1: The home limit switch is detected. 2: The home limit switch is cleared. 3: The home position During the sequence: 1. The axis moves in the Home Direction at the Home Speed to the home limit switch and stops. 2. The axis reverses direction and moves at the Home Return Speed until it clears the home limit switch and then stops. 3. The axis moves back to the home limit switch or it moves to the Offset position. The axis moves at the Home Return Speed. If the axis is a Rotary Axis, the move back to the Home Position takes the shortest path (that is, no more than ½ revolution). If the axis is past the home limit switch at the start of the homing sequence, the axis reverses direction and starts the return leg of the homing sequence. Use a Home Return Speed that is slower than the Home Speed to increase the homing accuracy. The accuracy of this sequence depends on the return speed and the delay to detect the transition of the home limit switch. Uncertainty = Home Return Speed x delay to detect the home limit switch. **Example**: Suppose your Home Return Speed is 0.1 in./s and it takes 10 ms to detect the home limit switch. Uncertainty = 0.1 in./s x 0.01 s = 0.001 in. The mechanical uncertainty of the home limit switch also affects the homing accuracy.

Sequence	Description
Active home to marker in forward bidirectional	The marker homing sequence is useful for single-turn rotary and linear encoder applications because these applications have only one encoder marker for full axis travel.



1: The encoder marker is detected.
2: The home position

During the sequence:

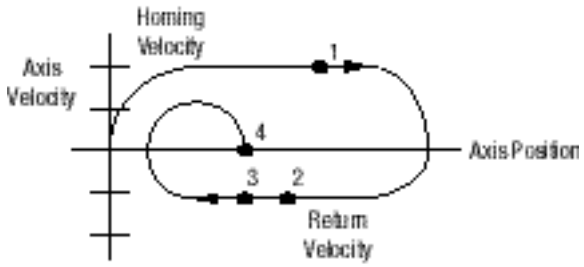
1. The axis moves in the Home Direction at the Home Speed to the marker and stops.
2. The axis moves back to the marker or it moves to the Offset position. The axis moves at the Home Return Speed. If the axis is a Rotary Axis, the move back to the Home Position takes the shortest path (that is, no more than ½ revolution).

The accuracy of this homing sequence depends on the homing speed and the delay to detect the marker transition.

Uncertainty = Home Speed x delay to detect the marker.

Example: Suppose your Home Speed is 1 in./s and it takes 1 μs to detect the marker.

Uncertainty = 1 In./s x 0.000001 s = 0.000001 in.

Sequence	Description
Active home to switch and marker in forward bidirectional	This is the most precise active homing sequence available.  1: The home limit switch is detected. 2: The home limit switch is cleared. 3: The encoder marker is detected. 4: The home position During the sequence: 1. The axis moves in the Home Direction at the Home Speed to the home limit switch and stops. 2. The axis reverses direction and moves at the Home Return Speed until it clears the home limit switch. 3. The axis keeps moving at the Home Return Speed until it gets to the marker. 4. The axis moves back to the marker or it moves to the Offset position. The axis moves at the Home Return Speed. If the axis is a Rotary Axis, the move back to the Home Position takes the shortest path (that is, no more than ½ revolution). If the axis is past the home limit switch at the start of the homing sequence, the axis reverses direction and starts the return leg of the homing sequence.
Active home to switch in forward unidirectional	This active homing sequence is useful for when an encoder marker is not available and either unidirectional motion is required or proximity switch is being used. During the sequence: 1. The axis moves in the Home Direction at the Home Speed to the home limit switch. 2. The axis moves to the Home Offset position if it's in the same direction as the Home Direction.
Active home to marker in forward unidirectional	This active homing sequence is useful for single-turn rotary and linear encoder applications when unidirectional motion is required. During the sequence: 1. The axis moves in the Home Direction at the Home Speed to the marker. 2. The axis moves to the Home Offset position if it's in the same direction as the Home Direction.

Sequence	Description
Active home to switch and marker in forward unidirectional	This active homing sequence is useful for multi-turn rotary applications when unidirectional motion is required. During the sequence: 1. The axis moves in the Home Direction at the Home Speed to the home limit switch. 2. The axis keeps moving at the Home Speed until it gets to the marker. 3. The axis moves to the Home Offset position if it's in the same direction as the Home Direction.

Passive Homing

Sequence	Description
Passive Immediate Home	This is the simplest passive homing sequence type. When this sequence is performed, the controller immediately assigns the Home Position to the current axis actual position. This homing sequence produces no axis motion.
Passive Home with Switch	This passive homing sequence is useful for when an encoder marker is not available or a proximity switch is being used. When this sequence is performed in the Passive Homing Mode, an external agent moves the axis until the home switch is detected. The Home Position is assigned to the axis position at the moment that the limit switch is detected. If you are using a Home Offset, then the Home Position is offset from the point where the switch is detected by this value.
Passive Home with Marker	This passive homing sequence is useful for single-turn rotary and linear encoder applications. When this sequence is performed in the Passive Homing Mode, an external agent moves the axis until the marker is detected. The home position is assigned to the axis position at the precise position where the marker was detected. If you are using a Home Offset, then the Home Position is offset from the point where the switch is detected by this value.
Passive Home with Switch then Marker	This passive homing sequence is useful for multi-turn rotary applications. When this sequence is performed in the Passive Homing Mode, an external agent moves the axis until the home switch and then the first encoder marker is detected. The home position is assigned to the axis position at the precise position where the marker was detected. If you are using a Home Offset, then the Home Position is offset from the point where the switch is detected by this value.

Notes:

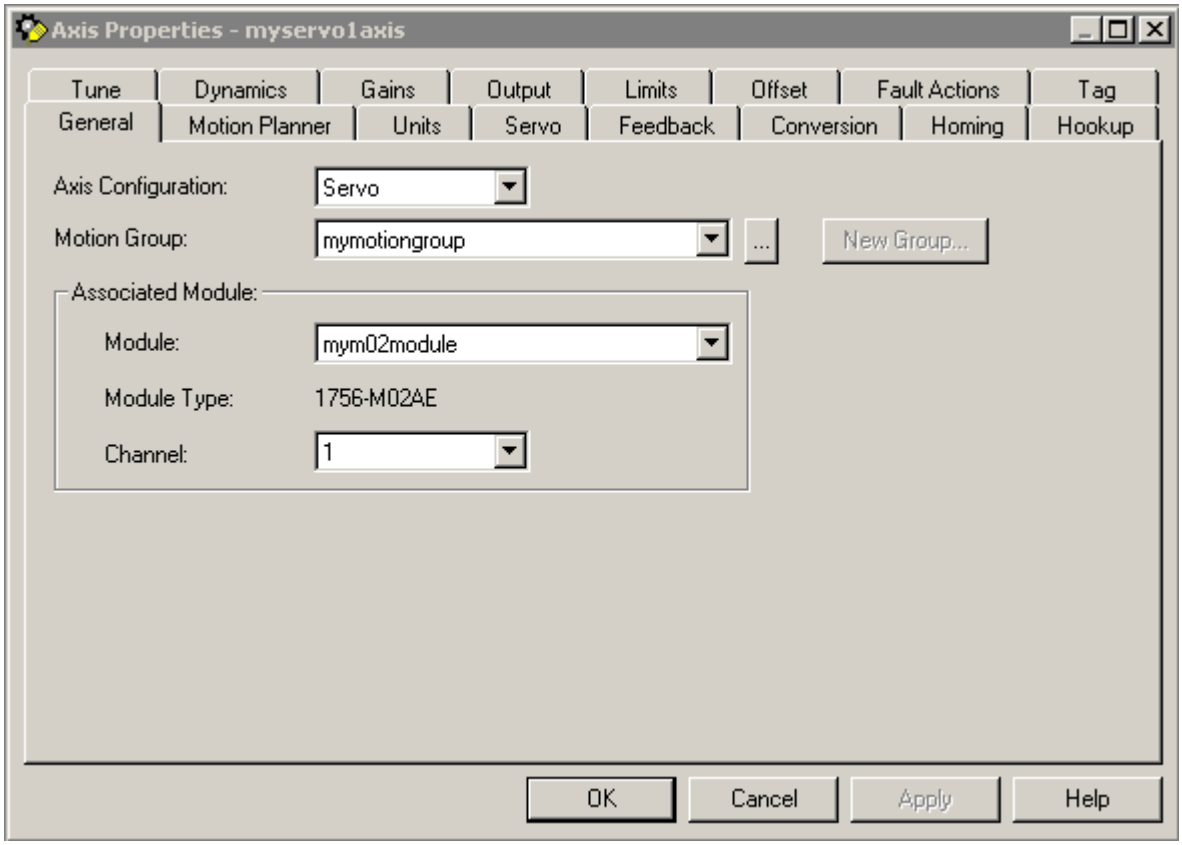
Axis Properties

Introduction

Use this appendix for a description of the properties of an axis.

General Tab – AXIS_SERVO

The General screen depicted below is for an AXIS_SERVO data type.



Axis Configuration

Selects and displays the intended use of the axis:

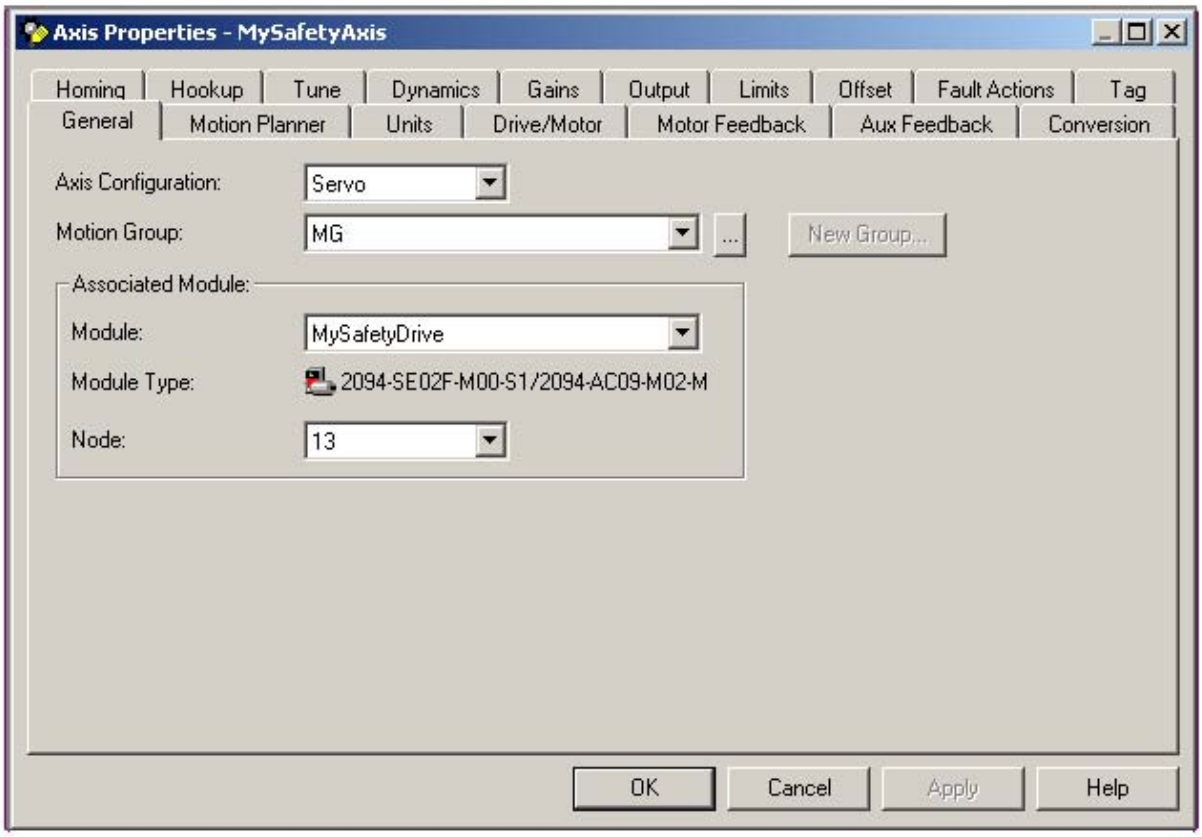
- Feedback Only: If the axis is to be used only to display position information from the feedback interface. This selection minimizes the display of axis properties tabs and parameters. The tabs for Servo, Tune, Dynamics, Gains, Output, Limits, and Offset are not displayed.
- Servo: If the axis is to be used for full servo operation. This selection maximizes the display of axis properties tabs and parameters.

Module Selects and displays the name of the motion module to which the axis is associated. Displays <none> if the axis is not associated with any motion module.

Channel Selects and displays the 1756-M02AE motion module channel - either 0 or 1 - to which the axis is assigned. Disabled when the axis is not associated with any motion module.

General Tab - AXIS_SERVO_DRIVE

The General screen shown below is for an AXIS_SERVO DRIVE Data Type.



Axis Configuration Selects and displays the intended use of the axis:

- Feedback Only: If the axis is to be used only to display position information from the feedback interface. This selection minimizes the display of axis properties tabs and parameters. The tabs for Tune, Dynamics, Gains, Output, Limits, and Offset are not displayed.

- Servo: If the axis is to be used for full servo operation. This selection maximizes the display of axis properties tabs and parameters.

Motion Group

Selects and displays the Motion Group to which the axis is associated. An axis assigned to a Motion Group appears in the Motion Groups branch of the Controller Organizer, under the selected Motion Group sub-branch. Selecting <none> terminates the Motion Group association, and moves the axis to the Ungrouped Axes sub-branch of the Motions Groups branch.

Module

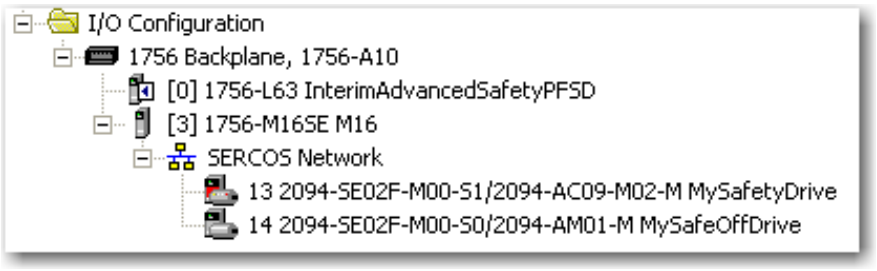
Selects and displays the name of the SERCOS drive to which the axis is associated. Displays <none> if the axis is not associated with any drive.

Module Type

Displays a module icon and the name of the SERCOS drive to which the axis is associated. Displays <none> if the axis is not associated with any drive. If the associated drive is a Kinetix Safety drive, a portion the module icon is red in color to signify its safety significance.

Following is a sample of thhe module icon (from the I/O configuration folder in RSLogix 5000) for:

- Kinetix 6000 Advanced Safety Drive (S1)
- Kinetix 6000 Enhanced Safe Torque-Off Drive (S0)



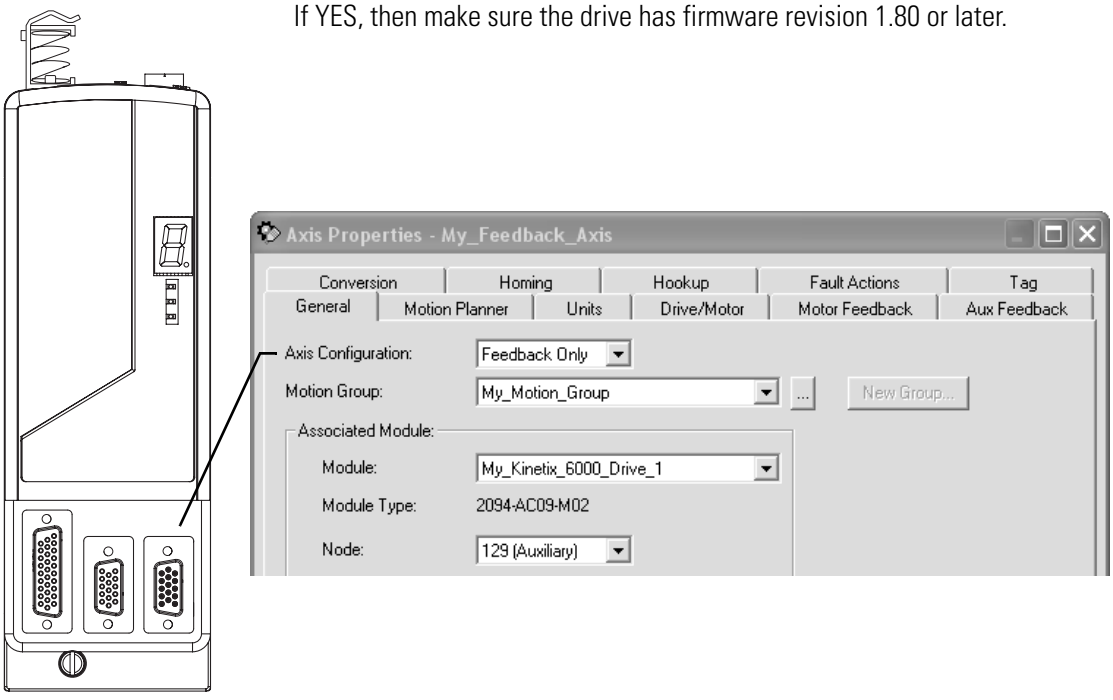
Node

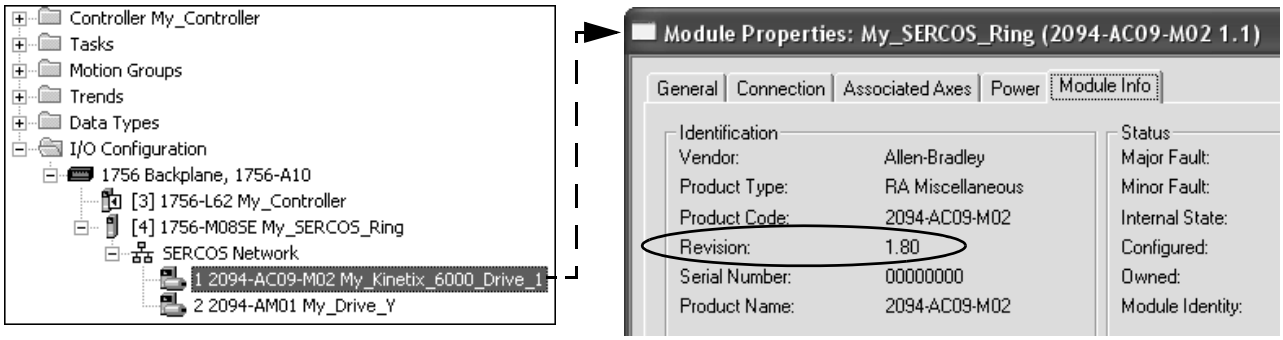
Displays the base node of the associated SERCOS drive. Disabled when the axis is not associated with any drive.

Node with a Kinetix 6000 Drive

IMPORTANT Do you want to use the auxiliary feedback port of a Kinetix 6000 drive as a feedback-only axis?

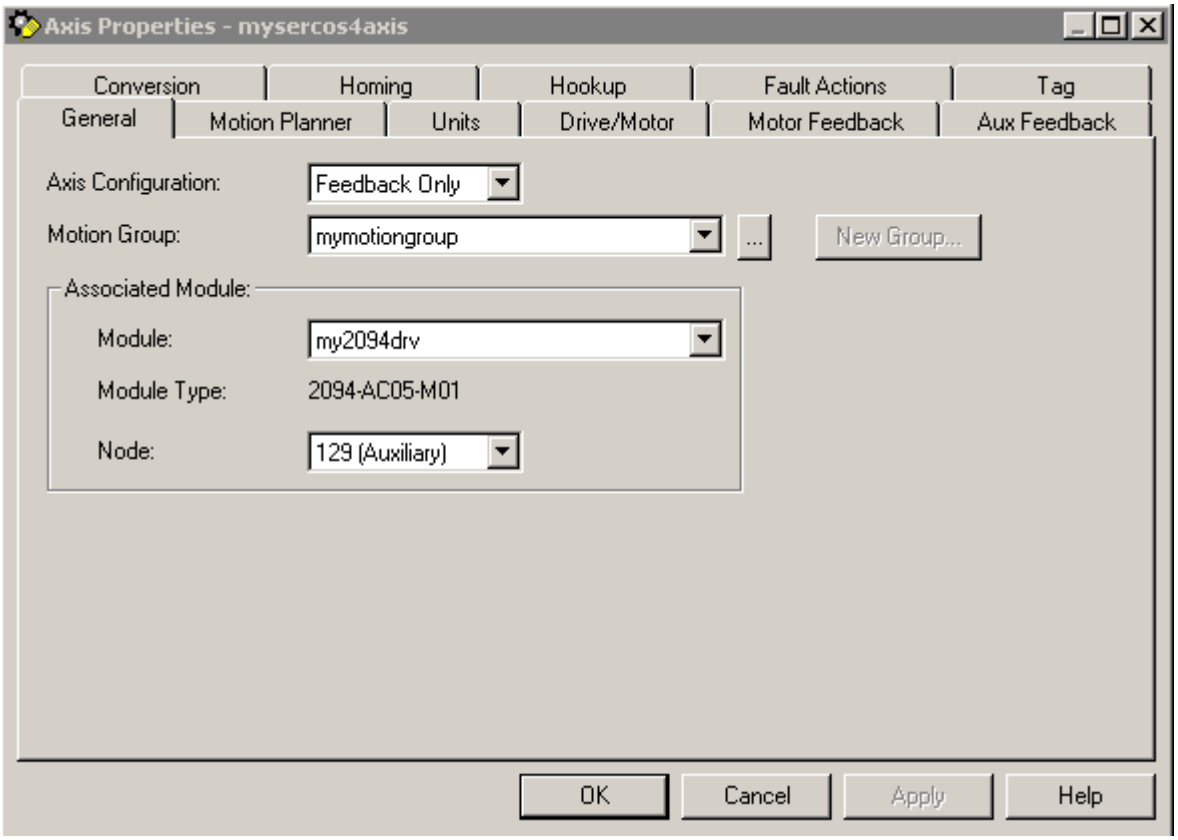
If YES, then make sure the drive has firmware revision 1.80 or later.





When a Kinetix 6000 drive is designated in the Associated Module box, there is an additional option for the Node value. It is the node associated with the drive plus 128 with (Auxiliary) after the number. The range is 129 to 234. When the Auxiliary Node assignment is chosen the axis configuration is

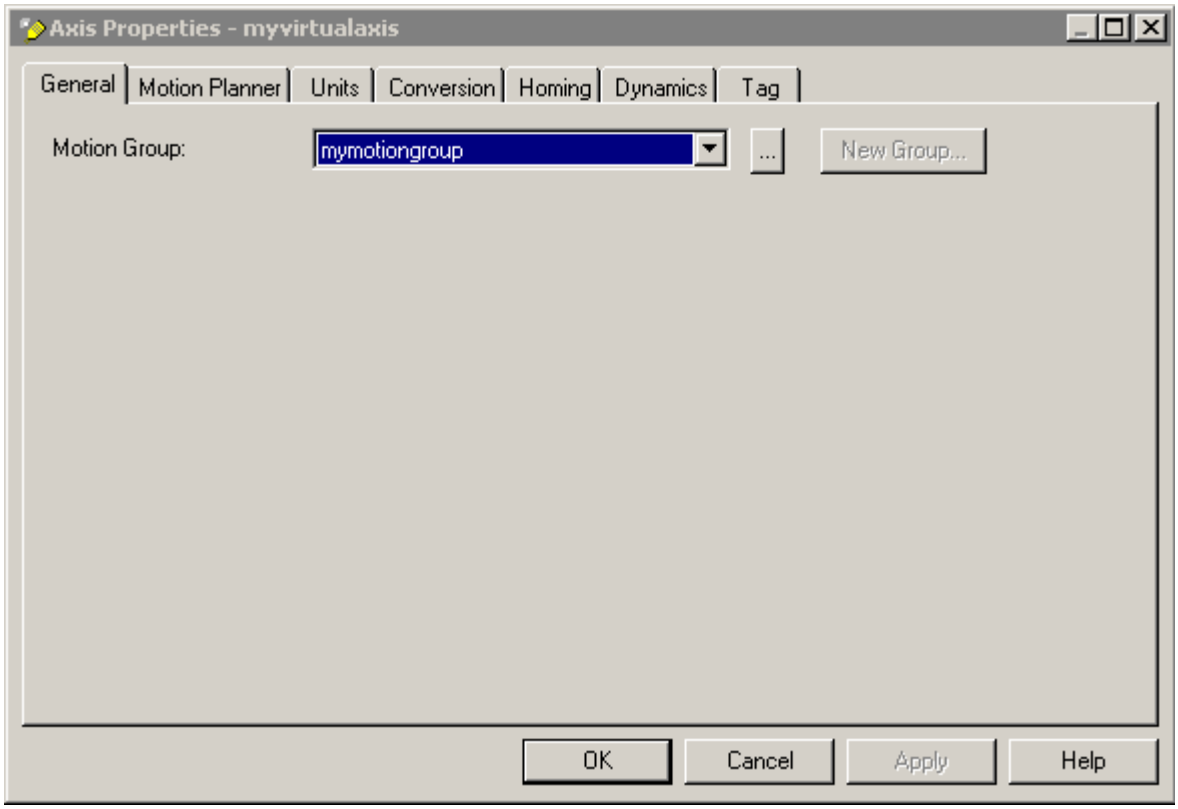
changed to Feedback Only on the General tab and the spat (*) appears next to General.



This also places a spat (*) on the Aux Feedback tab and you must go there and select the appropriate values. On the Drive/Motor tab the Loop Configuration is changed to Aux Feedback Only.

General Tab - AXIS_VIRTUAL

The AXIS_VIRTUAL General tab is shown below.

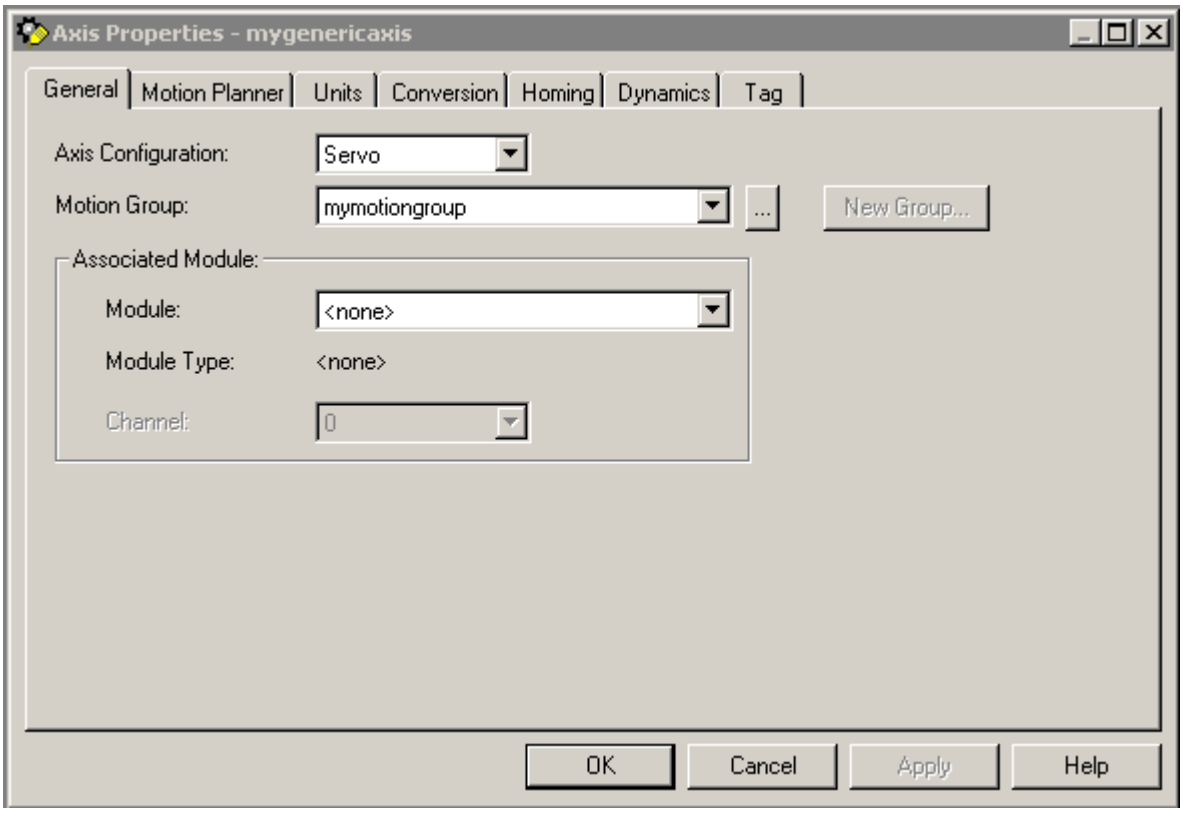


Motion Group

Selects and displays the Motion Group to which the axis is associated. An axis assigned to a Motion Group appears in the Motion Groups branch of the Controller Organizer, under the selected Motion Group sub-branch. Selecting <none> terminates the Motion Group association, and moves the axis to the Ungrouped Axes sub-branch of the Motions Groups branch.

General Tab – AXIS_GENERIC

The AXIS_GENERIC General tab is shown below.



Axis Configuration

Selects and displays the intended use of the axis:

- Feedback Only: If the axis is to be used only to display position information from the feedback interface. This selection minimizes the display of axis properties tabs and parameters. The tab for Dynamics is not available.
- Servo: If the axis is to be used for full servo operation. This selection maximizes the display of axis properties tabs and parameters.

Motion Group

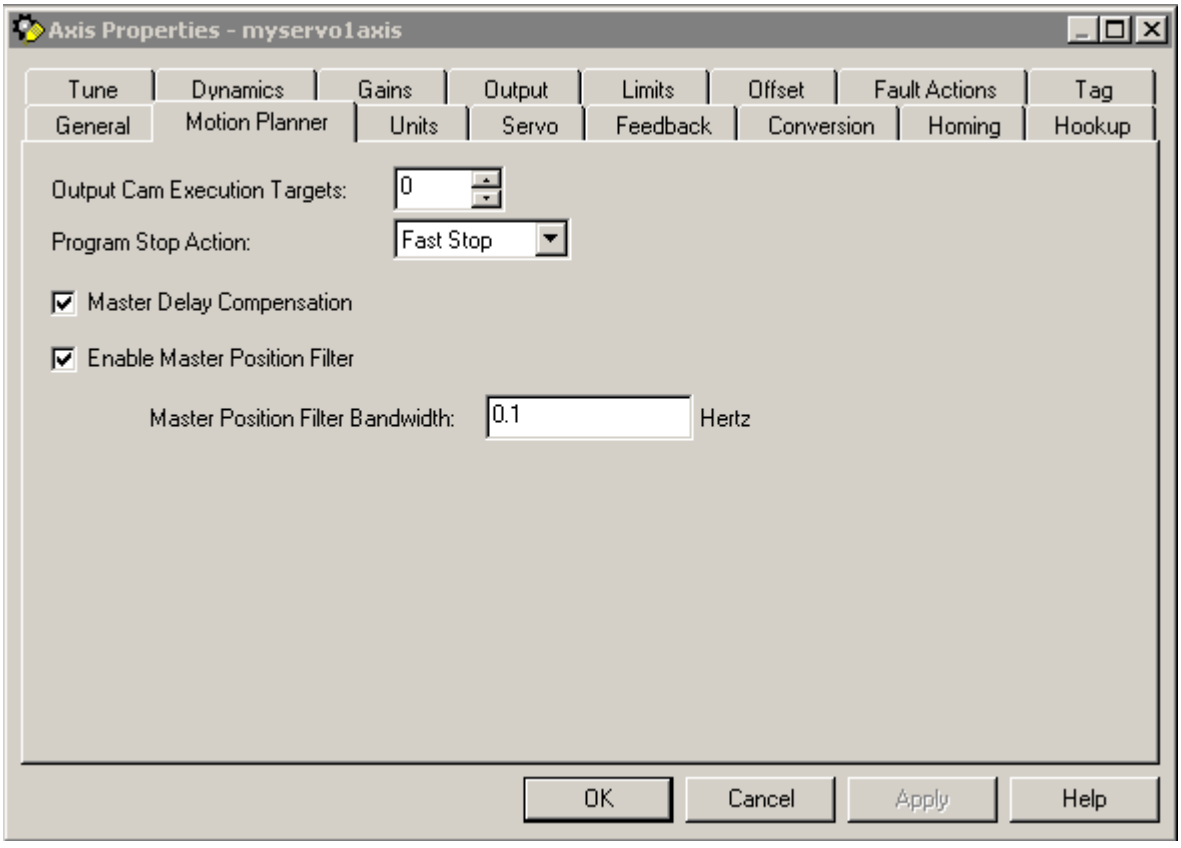
Selects and displays the Motion Group to which the axis is associated. An axis assigned to a Motion Group appears in the Motion Groups branch of the Controller Organizer, under the selected Motion Group sub-branch. Selecting <none> terminates the Motion Group association, and moves the axis to the Ungrouped Axes sub-branch of the Motions Groups branch.

Module Selects and displays the name of the motion module to which the axis is associated. Displays <none> if the axis is not associated with any motion module.

Channel Selects and displays the motion module channel - either 0 or 1 - to which the axis is assigned. Disabled when the axis is not associated with any motion module.

Motion Planner Tab

The Motion Planner tab is where you set/edit the number of Output Cam execution targets, the type of stop action to use, enable or disable Master Delay Compensation, enable or disable Master Position Filter, and set the bandwidth for Master Position Filter Bandwidth. The Motion Planner tab has the same fields regardless of the type of axis.



Output Cam Execution Targets Determines how many Output Cam execution nodes (instances) are created for a specific axis. Note that the Execution Target parameter for the MAOC/MDOC instructions specify which of the configured execution nodes the instruction is affecting. In addition, the number specified in the Axis

Properties dialog specifies the number of instances of Output Cam in which the value of zero means “none”, and the value specified for Execution Target in the MAOC instruction references a specific instance in which a value of zero selects the first instance.

Program Stop Action

Select how a specific axis is stopped when the processor undergoes a mode change, or when an explicit Motion Group Programmed Stop (MGPS) instruction is executed.

- **Fast Disable:** The axis is decelerated to a stop using the current configured value for maximum deceleration. Servo action is maintained until the axis motion has stopped at which time the axis is disabled (that is, Drive Enable is disabled, and Servo Action is disabled).
- **Fast Shutdown:** The axis is decelerated to a stop using the current configured value for maximum deceleration. Once the axis motion is stopped, the axis is placed in the shutdown state (that is, Drive Enable is disabled, Servo Action is disabled, and the OK contact is opened). To recover from this state, a reset instruction must be executed.
- **Fast Stop:** The axis is decelerated to a stop using the current configured value for maximum deceleration. Servo action is maintained after the axis motion has stopped. This mode is useful for gravity or loaded systems, where servo control is needed at all times.
- **Hard Disable:** The axis is immediately disabled (that is, Drive Enable is disabled, Servo Action is disabled, but the OK contact is left closed). Unless the drive is configured to provide some form of dynamic breaking, this results in the axis coasting to a stop.
- **Hard Shutdown:** The axis is immediately placed in the shutdown state. Unless the drive is configured to provide some form of dynamic breaking, this results in the axis coasting to a stop. To recover from this state, a reset instruction must be executed.

Master Delay Compensation Checkbox

Use this checkbox to Enable/Disable Master Delay Compensation. Master Delay Compensation is used balance the delay time between reading the master axis command position and applying the associated slave command position to the slave’s servo loop. This feature ensures that the slave axis command position accurately tracks the actual position of the master axis that is, zero tracking error.

Clicking on this box enables Master Delay Compensation. The default setting is Disabled.

If the axis is configured for Feedback only, Master Delay Compensation should be disabled.

Enable Master Position Filter Checkbox

Use this checkbox to Enable/Disable Master Position Filter. The default is disabled and must be checked to enable position filtering. Master Position Filter, when enabled, effectively filters the specified master axis position input to the slave axis's gearing or position camming operation. The filter smooths out the actual position signal from the master axis, and thus smooths out the corresponding motion of the slave axis.

When this feature is enabled the Master Position Filter Bandwidth field is enabled.

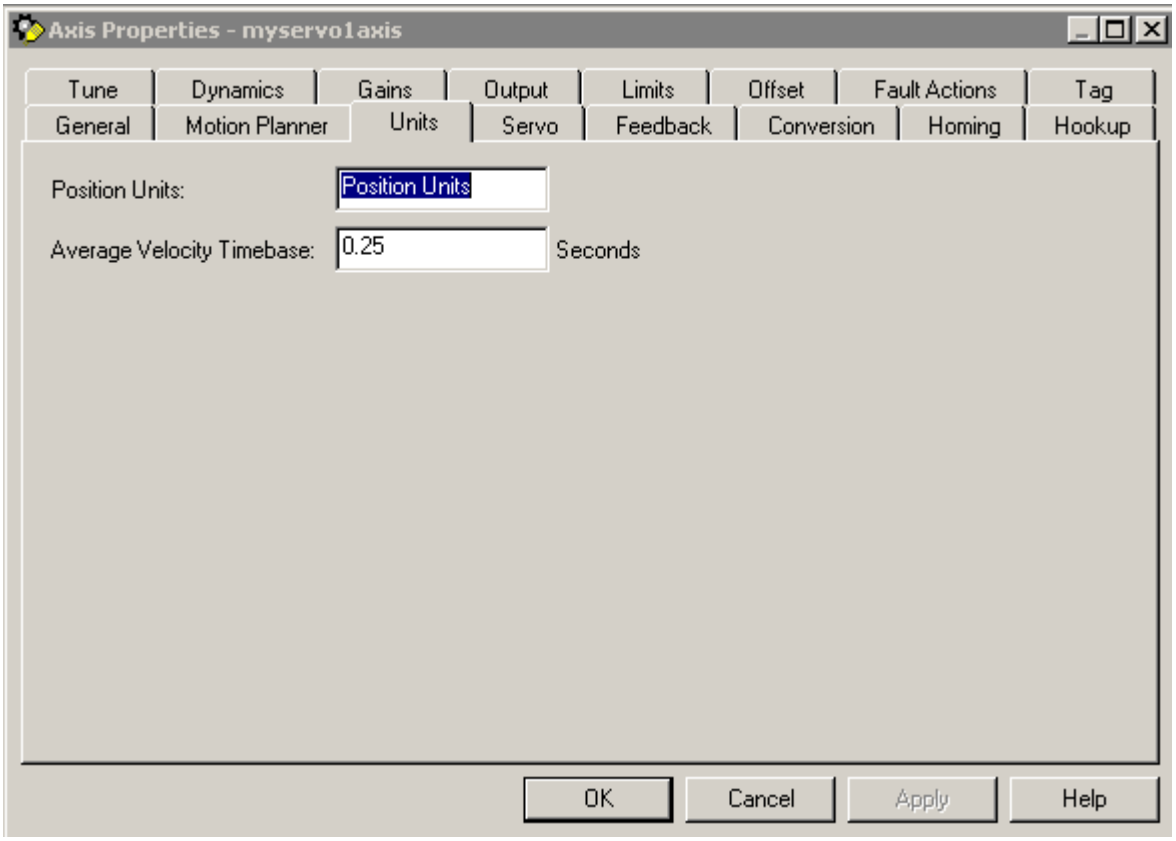
Master Position Filter Bandwidth

The Master Position Filter Bandwidth field is enabled when the Enable Position Filter checkbox is selected. This field controls the bandwidth for master position filtering. Enter a value in Hz in this field to set the bandwidth to for the Master Position Filter.

IMPORTANT A value of zero for Master Position Filter Bandwidth effectively disables the master position filtering.

Units Tab

The Units tab is the same for all axis data types. Use this tab to determine the units to define your motion axis.



Position Units

User-defined engineering units (rather than feedback counts) used for labeling all motion-related values (for example, position, velocity, and so on) These position units can be different for each axis.

Position Units should be chosen for maximum ease of use in your application. For example, linear axes might use position units of Inches, Meters, or mm whereas rotary axes might use units of Revs or Degrees.

Average Velocity Timebase

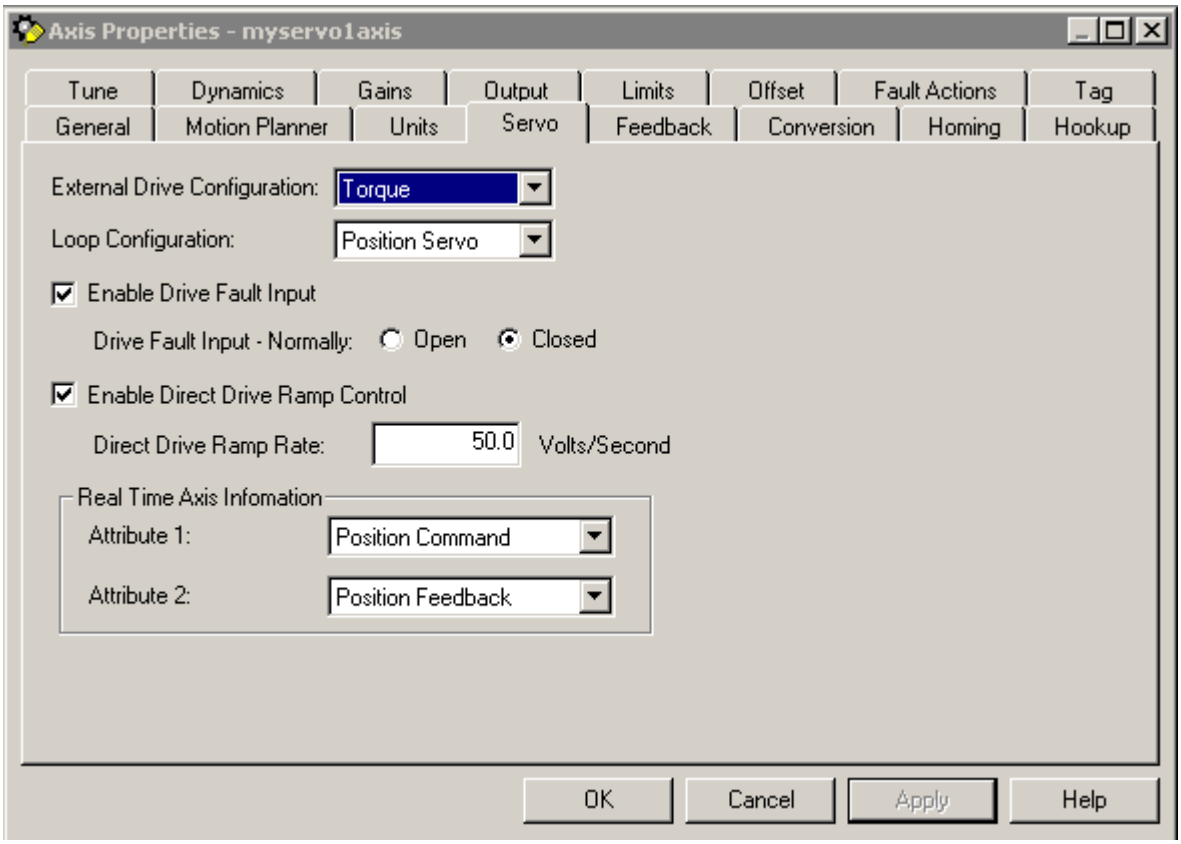
Specifies the time (in seconds) to be used for calculating the average velocity of the axis. This value is computed by taking the total distance the axis travels in the amount of time specified, and dividing this value by the timebase.

The average velocity timebase value should be large enough to filter out the small changes in velocity that would result in a noisy velocity value, but small enough to track significant changes in axis velocity. A value of 0.25 to 0.50 seconds should work well for most applications.

Click on the Apply button to accept your changes.

Servo Tab - AXIS_SERVO

Click on the Servo tab from the Axis Properties for AXIS_SERVO to access the Servo dialog.



External Drive Configuration

Select the drive type for the servo loop:

- Velocity - disables the servo module's internal digital velocity loop.
- Torque - the servo module's internal digital velocity loop is active, which is the required configuration for interfacing the servo axis to a torque loop servo drive.
- Hydraulic - enables features specific to hydraulic servo applications.

Loop Configuration

Select the configuration of the servo loop. For this release, only Position Servo is available.

Enable Drive Fault Input

Check this box if you wish to enable the Drive Fault Input. When active the motion module receives notice whenever the external drive detects a fault.

Drive Fault Input

Specifies the usual state of the drive fault input when a fault is detected on the drive.

- Normally Open – when a drive fault is detected it opens its drive fault output contacts.
- Normally Closed – when a drive fault is detected it closes its drive fault output contacts.

Enable Direct Drive Ramp Control

Clicking on the Enable Direct drive Ramp Control check box lets you set the Direct Drive Ramp Rate in volts per second for when an MDO instruction is executed.

Direct Drive Ramp Rate

The Direct Drive Ramp Rate is a slew rate for changing the output voltage when a Direct Drive On (MDO) instruction is executed. A Direct Drive Ramp Rate of 0 disables the output rate limiter letting the Direct Drive On voltage to be applied directly.

Real Time Axis Information

Attribute 1/Attribute 2

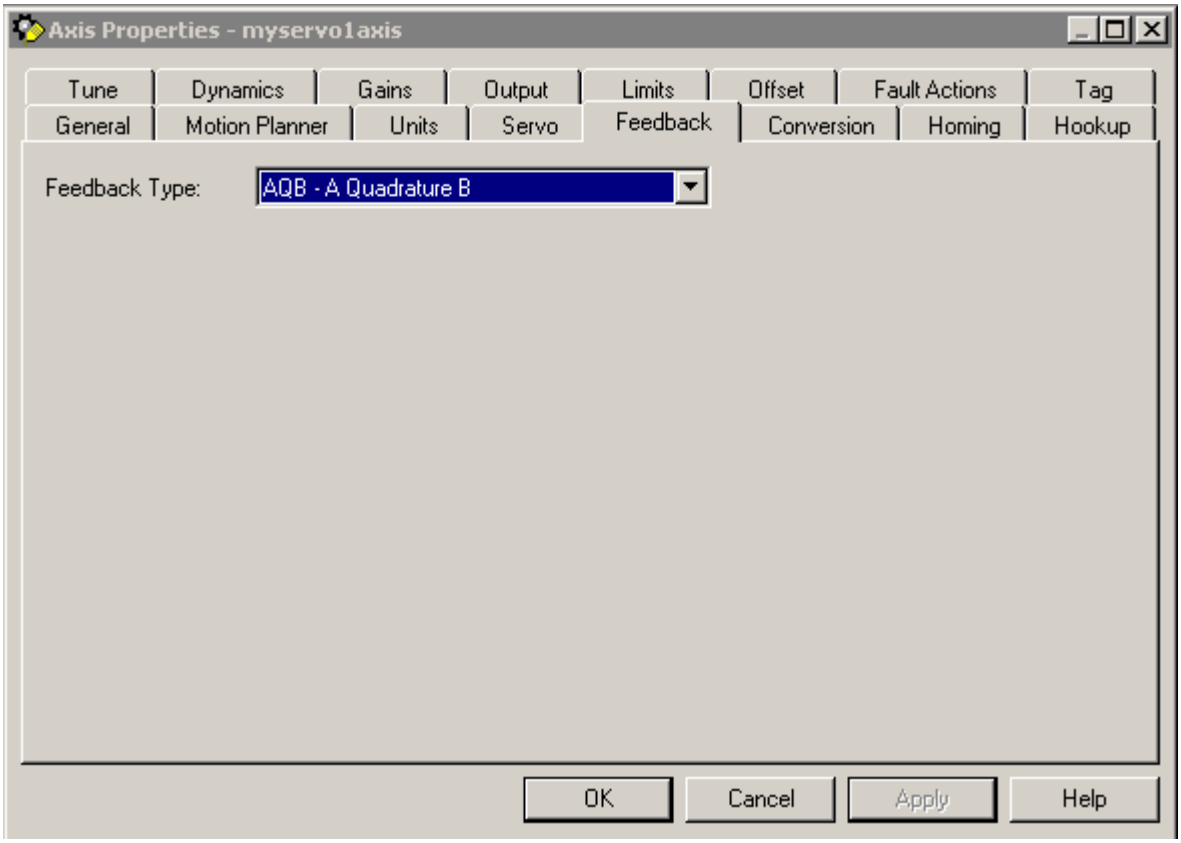
Select up to two axis attributes whose status are transmitted – along with the actual position data – to the Logix processor. The values of the selected attributes can be accessed via the standard GSV or Get Attribute List service.

The servo status data update time is precisely the coarse update period.

If a GSV is done to one of these servo status attributes without having selected this attribute via the Drive Info Select attribute, the attribute value is static and does not reflect the true value in the servo module.

Feedback Tab – (AXIS_SERVO)

The Feedback tab lets you to select the type of Feedback used with your Servo axis.



Feedback Type

Select the appropriate Feedback for your current configuration. Your options are dependent upon the motion module to which the axis is associated.

A Quadrature B Encoder Interface (AQB)

The 1756-M02AE servo module provides interface hardware to support incremental quadrature encoders equipped with standard 5-Volt differential encoder-interface signals. The AQB option has no associated attributes to configure.

Synchronous Serial Interface (SSI)

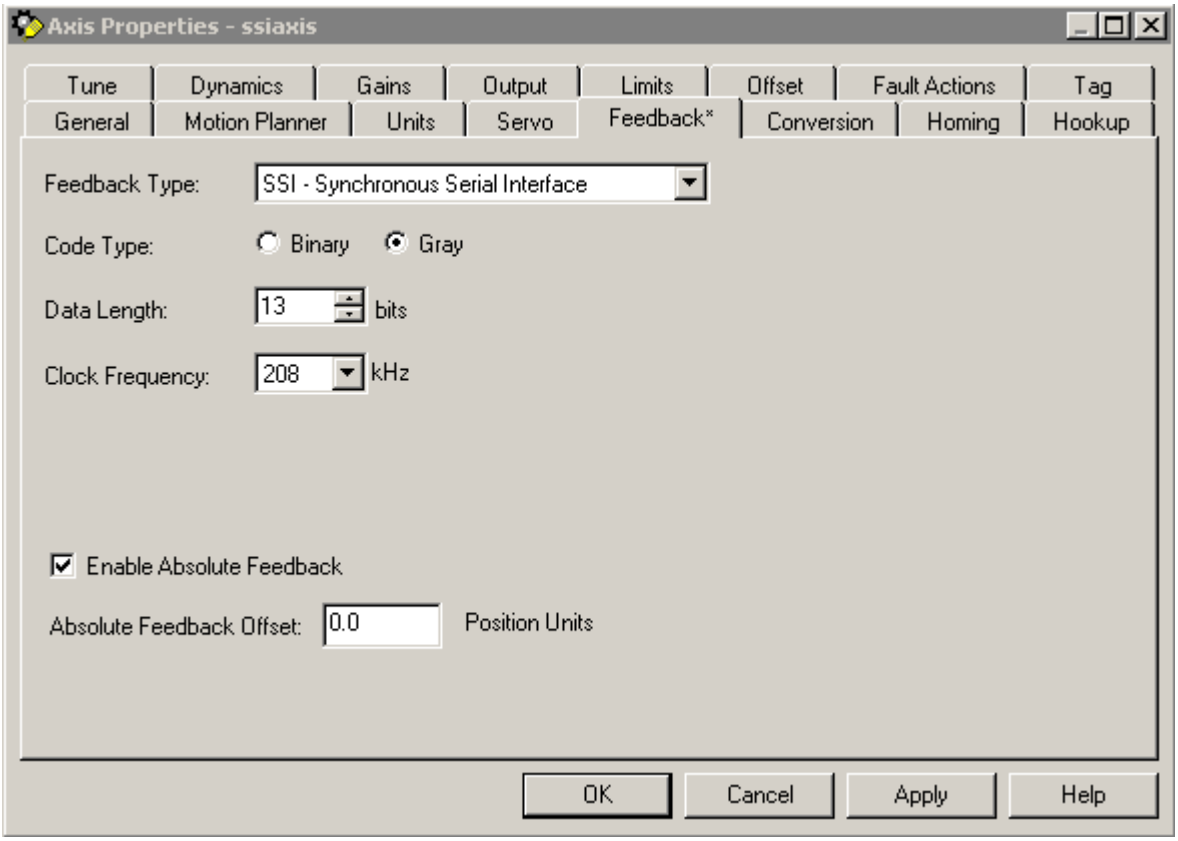
The 1756-M02AS servo module provides an interface to transducers with Synchronous Serial Interface (SSI) outputs. SSI outputs use standard 5V differential signals (RS422) to transmit information from the transducer to the controller. The signals consist of a Clock generated by the controller and Data generated by the transducer.

Linear Displacement Transducer (LDT)

The 1756-HYD02 Servo module provides an interface to the Linear Magnetostrictive Displacement Transducer, or LDT. A Field Programmable Gate Array (FPGA) is used to implement a multi-channel LDT Interface. Each channel is functionally equivalent and is capable of interfacing to an LDT device with a maximum count of 240,000. The LDT interface has transducer failure detection and digital filtering to reduce electrical noise.

The Feedback screen changes in appearance depending on the selected Feedback Type.

When the servo axis is associated with a 1756-M02AS motion module the only Feedback Type available is SSI-Synchronous Serial Interface and the Feedback tab dialog looks like the following illustration.



Code Type

The type of code, either Binary or Gray, used to report SSI output. If the module's setting does not match the feedback device, the positions jump around erratically as the axis moves.

Data Length

The length of output data in a specified number of bits between 8 and 31. The data length for the selected feedback device can be found in its specifications.

Clock Frequency

Sets the clock frequency of the SSI device to either 208 (default) or 625 kHz. When the higher clock frequency is used, the data from the feedback device is more recent, but the length of the cable to the transducer must be shorter than with the lower frequency.

Enable Absolute Feedback

This checkbox allows you to either enable (checked) or disable (unchecked) the Absolute Feedback feature. The default is enabled. If Enable Absolute Feedback is set, the servo module adds the Absolute Feedback Offset to the current position of the feedback device to establish the absolute machine reference position. Absolute feedback devices retain their position reference even through a power-cycle, therefore the machine reference system can be restored at power-up.

Absolute Feedback Offset

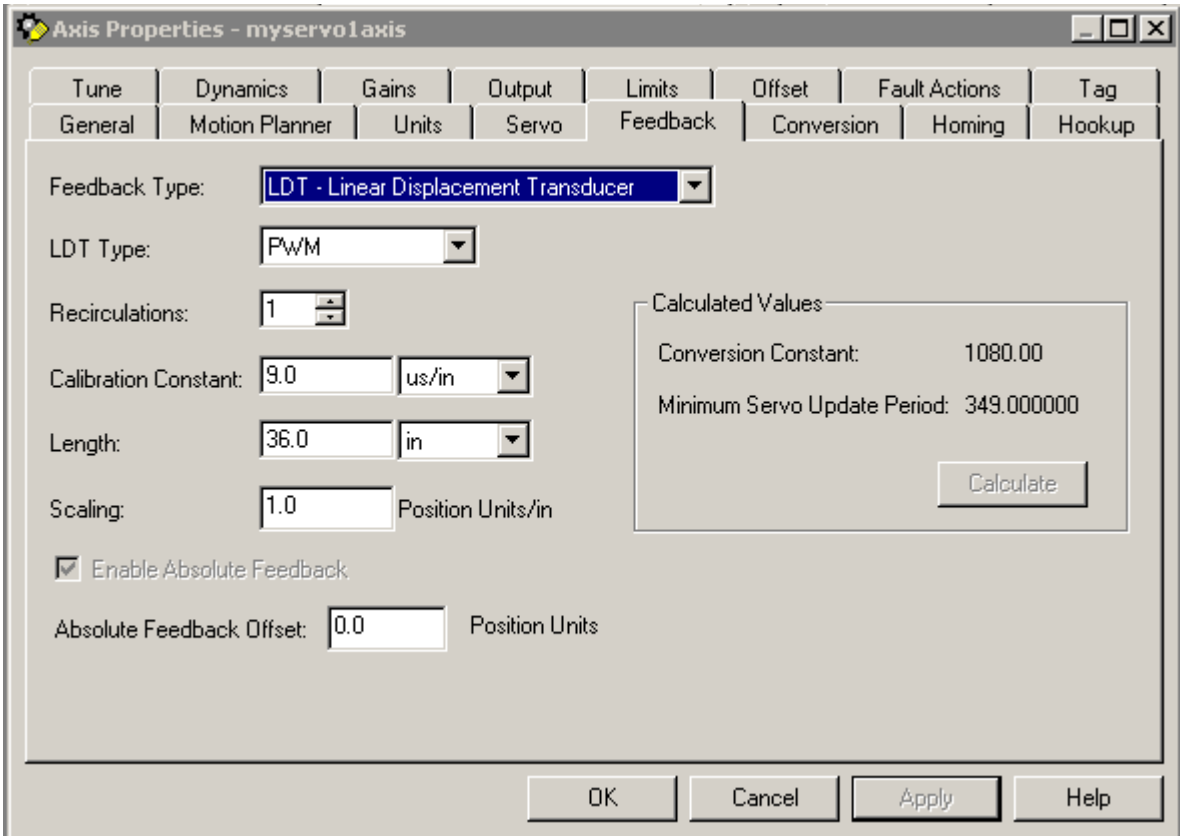
If Absolute feedback is enabled, this field becomes active. You can enter the amount of offset, in position units, to be added to the current position of the Feedback device.

The SSI is an absolute feedback device. To establish an appropriate value for the Offset, the MAH instruction can be executed with the Home Mode set to Absolute (the only valid option if Enable Absolute Feedback is enabled). When executed, the module computes the Absolute Feedback Offset as the difference between the configured value for Home Position and the current absolute feedback position of the axis. The computed Absolute Feedback Offset is immediately applied to the axis upon completion of the MAH instruction. The actual position of the axis is re-referenced during execution of the MAH instruction therefore, the servo loop must not be active. If the servo loop is active, the MAH instruction errors.

When the Enable Absolute Feedback is disabled, the servo module ignores the Absolute Feedback Offset and treats the feedback device as an incremental position transducer. A homing or redefine position operation is required to establish the absolute machine reference position. The Absolute Home Mode is invalid.

If using Single-turn or Multi-turn Absolute SSI Feedback transducers, see the Homing tab information for important details concerning Absolute feedback tranducer's marker reference.

When the servo axis is associated to a 1756-HYD02 motion module, then LDT - Linear Displacement Transducer is the only option for Feedback Type.



LDT Type This field selects the type of LDT to use to provide feedback to the Hydraulic module. The available types are PWM, Start/Stop Rising, or Start/Stop Falling.

Recirculations Use this field to set the number of repetitions to use to acquire a measurement from an LDT.

Calibration Constant This is a number that is engraved on the LDT by the manufacturer. It specifies the characteristics of the individual LDT. Each LDT has its own calibration constant therefore, if you change the LDT, you must change the Calibration constant.

Length This value defines the stroke of travel of the hydraulic cylinder. The length value is used with the number of recirculations to determine the minimum servo update period.

Scaling

Scaling defines the relationship between the LDT unit of measure (length field) and the unit of measure defined at the Units tab.

Enable Absolute Feedback

This field is grayed out because it is always active when Feedback Type is LDT.

Absolute Feedback Offset

Enter the amount of offset, in position units, to be added to the current position of the LDT.

The LDT is an absolute feedback device. To establish an appropriate value for the Offset, the MAH instruction can be executed with the Home Mode set to Absolute (the only valid option if Enable Absolute Feedback is enabled). When executed, the module computes the Absolute Feedback Offset as the difference between the configured value for Home Position and the current absolute feedback position of the axis. The computed Absolute Feedback Offset is immediately applied to the axis upon completion of the MAH instruction. The actual position of the axis is re-referenced during execution of the MAH instruction therefore, the servo loop must not be active. If the servo loop is active, the MAH instruction errors.

When the Enable Absolute Feedback is disabled, the servo module ignores the Absolute Feedback Offset and treats the feedback device as an incremental position transducer. A homing or redefine position operation is required to establish the absolute machine reference position. The Absolute Home Mode is invalid.

Calculated Values

Conversion Constant

The Conversion Constant is calculated from the values entered on the Feedback screen when the Calculate button is selected. This calculated value must be typed into the Conversion Constant field on the Conversion tab as it is not automatically updated.

Minimum Servo Update Period

The Minimum Servo Update period is calculated based on the values entered for Recirculations and Length on the Feedback tab. When these values are changed, selecting the Calculate button recalculates the Minimum Servo Update Period based on the new values.

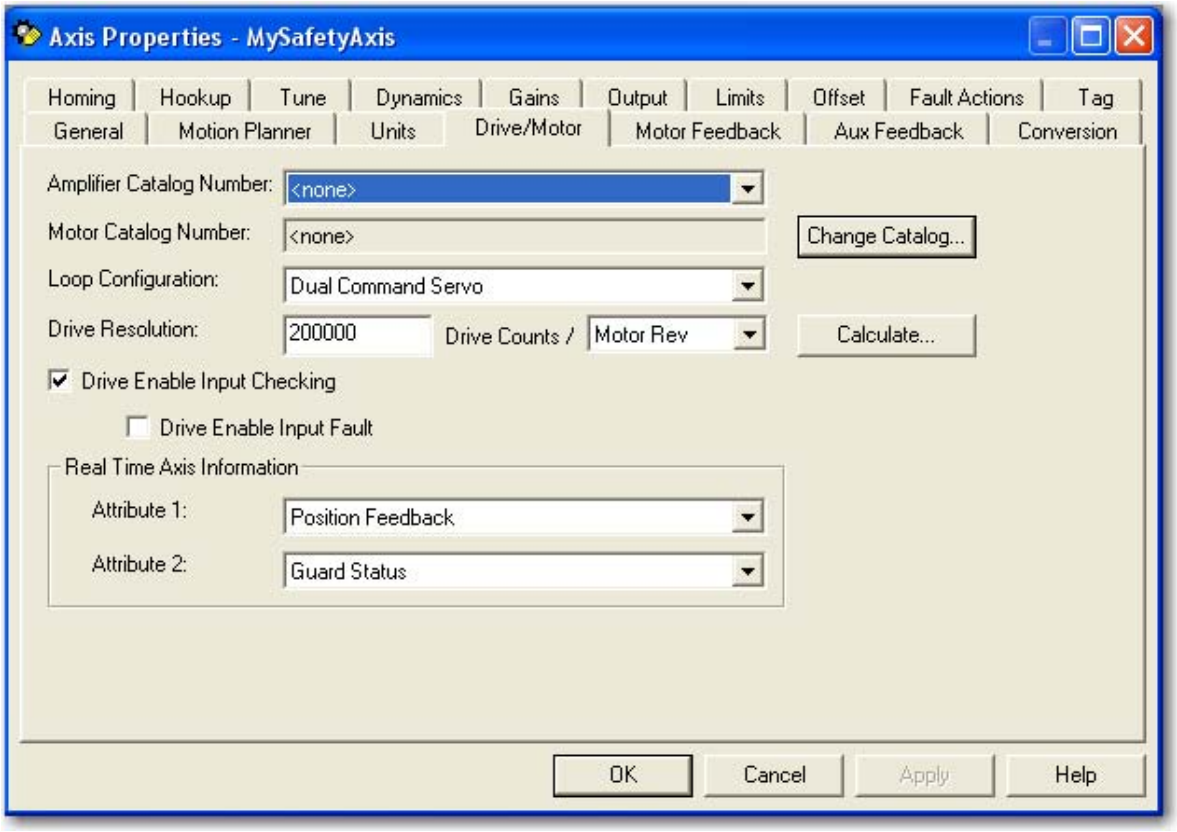
Calculate Button

The Calculate Button becomes active whenever you make changes to the values on the Feedback tab. Clicking on the Calculate Button recalculates the

Conversion Constant and Minimum Servo Update Period values. however, you must then reenter the Conversion Constant value at the Conversion tab as the values are not updated automatically.

Drive/Motor Tab - (AXIS_SERVO_DRIVE)

Use this tab to configure the servo loop for an AXIS_SERVO_DRIVE axis, and open the Change Catalog dialog box.



Amplifier Catalog Number

Select the catalog number of the amplifier to which this axis is connected.

Catalog Number

Select the catalog number of the motor associated with this axis. When you change a Motor Catalog Number, the controller recalculates the values of the

following values using (among other values) the default Damping Factor of 0.8.

On this tab or dialog	These attributes are recalculated
Motor Feedback tab	Motor Feedback Type Motor Feedback Resolution
Gains tab	Position Proportional Gains Velocity Proportional Gains
Dynamics tab	Maximum Velocity Maximum Acceleration Maximum Deceleration
Limits tab	Position Error Tolerance
Custom Stop Action Attributes dialog	Stopping Torque
Custom Limit Attributes dialog	Velocity Limit Bipolar Velocity Limit Positive Velocity Limit Negative Acceleration Limit Bipolar Acceleration Limit Positive Acceleration Limit Negative Torque Limit Bipolar Torque Limit Positive Torque Limit
Tune Bandwidth dialog	Position Loop Bandwidth Velocity Loop Bandwidth

The Associated Module selection (selected on the General tab), determines available catalog numbers.

Loop Configuration

Select the configuration of the servo loop:

- Motor Feedback Only – Displayed when Axis Configuration is Feedback only
- Aux Feedback Only – Displayed when Axis Configuration is Feedback only
- Position Servo
- Aux Position Servo (not applicable to Ultra3000 drives)

- Dual Position Servo
- Dual Command Servo
- Aux Dual Command Servo
- Velocity Servo
- Torque Servo
- Dual Command/Feedback Servo

Drive Resolution

Type in the number of counts per motor revolution, motor inch, or motor millimeter. This value applies to all position data. Valid values range from 1 to 2^32 - 1. One Least Significant Bit (LSB) for position data equals 360° / Rotational Position Resolution.

Drive Resolution is also referred to as Rotational Position Resolution.

When you save an edited Drive Resolution value, a message box appears, asking you if you want the controller to automatically recalculate certain attribute settings.

Drive Resolution is especially helpful for either fractional unwind applications or multi-turn applications requiring cyclic compensation. You can modify the Drive Resolution value so that dividing it by the Unwind Value yields a whole integer value. The higher the Drive Resolution setting, the finer the resolution.

Drive Enable Input Checking

To activate Drive Enable Input Checking click on the checkbox. When active (box is checked) the drive regularly monitors the state of the Drive Enable Input. This dedicated input enables the drive's power structure and servo loop. If Drive Enable Input Checking is not active then no such checking of the Drive Enable Input occurs.

Drive Enable Input Fault

Click on the checkbox to activate the Drive Enable Input Fault. When active, a fault detected on the external drive notifies the motion module via Drive Fault Input.

Real Time Axis Information

Attribute 1/Attribute 2

Select up to two axis attributes whose status are transmitted – along with the actual position data – to the Logix processor. You can access the values of the selected attributes using a GSV command or from the axis tag itself. This data is transmitted at a rate equal to the servo status data update time.

If you issue a GSV command for servo status attribute, or use the value from the axis tag, without having selected this attribute via the Drive Info Select attribute, the attribute value is static and does not reflect the true value in the servo module.

If the AXIS_SERVO_DRIVE is associated with a Kinetix Enhanced Safe Torque-Off or Advanced Safety Drive, these two additional Real Time Axis attributes are available.

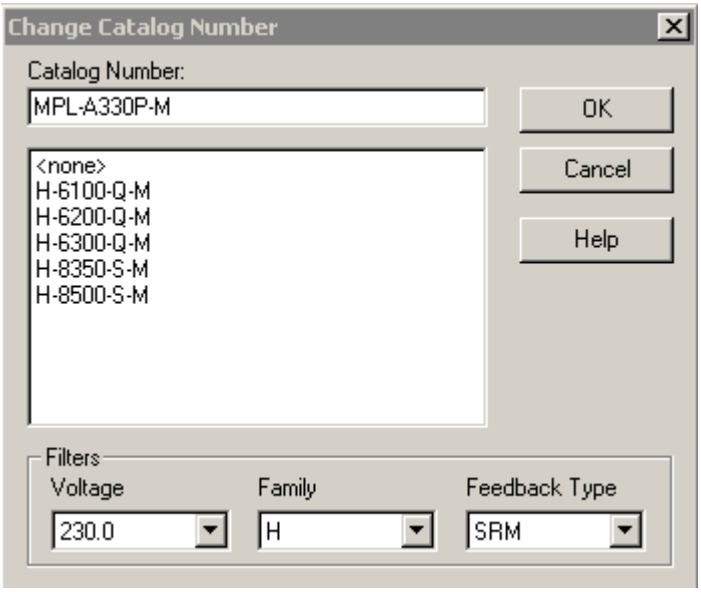
- Guard Status
- Guard Faults

If a AXIS_SERVO_DRIVE is associated with a Kinetix Advanced Safety Drive, we recommend that you configure the Guard Status attribute. Otherwise, you will receive a warning when verifying your project.

If the AXIS_SERVO_DRIVE is associated with a Kinetix Advanced Safety Drive, and	Then
Either Attribute 1 or Attribute 2 is populated as Guard Status	No action is taken
Neither Attribute 1 or Attribute 2 is populated as Guard Status	Attribute 2 is populated as Guard Status
Attribute 2 is populated with an attribute other than Guard Status, and Attribute 1 is undefined	Attribute 1 is populated as Guard Status
Both Attribute 1 and Attribute 2 are populated with an attribute other than Guard Status	Upon project verification, a warning is issued.

Change Catalog…button

The Change Catalog button accesses the motor database and provides for selecting a new motor catalog number. There are three boxes that can be used for refine the selection process.



Catalog Number

Lists the available catalog numbers from the Motor Database based on any selection criteria from the Filters fields.

Filters

There are three optional Filter fields that allow you to refine your search of the Motor Database. The Filter boxes are defaulted to all.

Voltage

Lets you select a voltage rating from the pull-down list to broaden or narrow your search. The default is all.

Family

The Family filter box pull down list lets you narrow your motor search by restricting it to a particular family of motors. The default is all.

Feedback Type

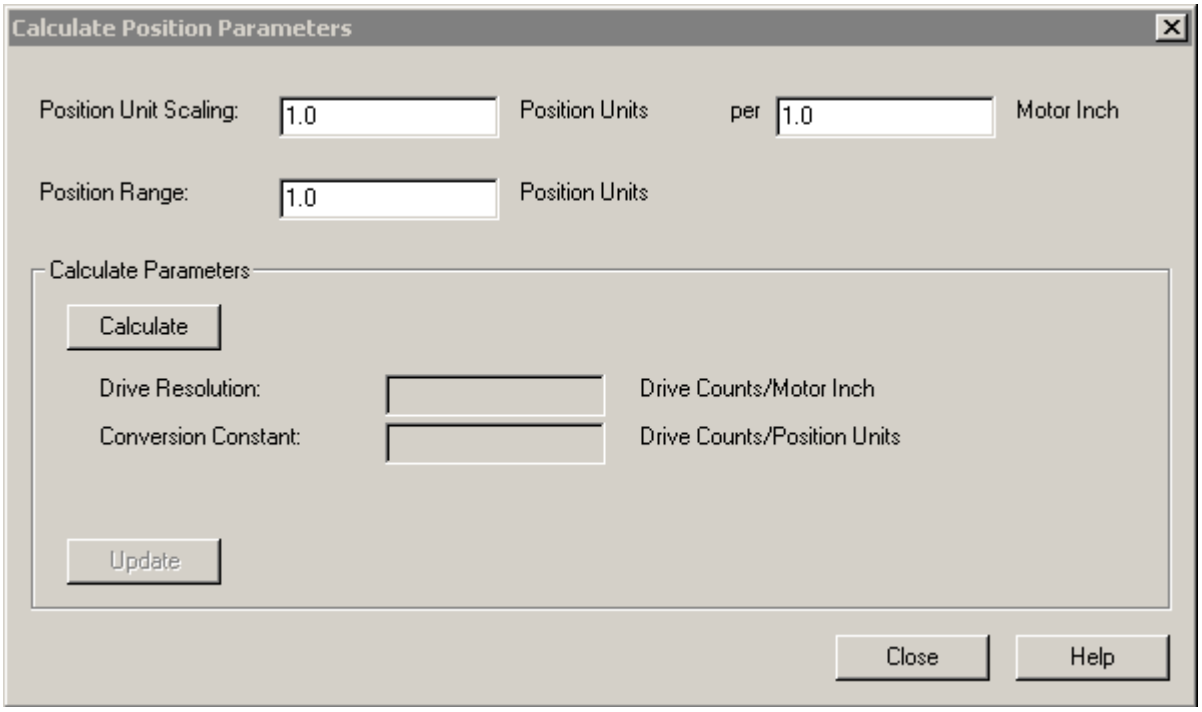
The Feedback Type filter box pull-down list lets you manipulate your motor search by acceptable Feedback types. The default is all.

Calculate... button

The Calculate button takes you to an input screen that is designed to calculate the Drive Resolution and Conversion Constant based upon your input for Position Unit Scaling and Position Range for Linear Positioning mode. If you

are in Rotary Positioning Mode then it calculates the Drive Resolution, Conversion Constant, and Position Unwind based upon your inputs for Position Unit Scaling and Position Unit Unwind.

When the Conversion screen has Linear as the value for Position Mode, clicking on the Calculate button displays the following screen.



Position Unit Scaling Position Unit Scaling defines the relationship between the Position Units defined on the Units tab and the units selected to measure position.

Per The units used for Position Unit Scaling. The options are: Motor Inch, Motor Millimeter, or Motor Rev

Position Range Maximum travel limit that your system can go.

Position Unit Unwind For Rotary applications, the Position Unit Unwind field displays. Enter the value for the maximum number of unwinds in position units per unwind cycle.

Calculate Parameters

The Calculate Parameters shows the values that are to be calculated based upon the values entered for the Position Unit Scaling and Position Range.

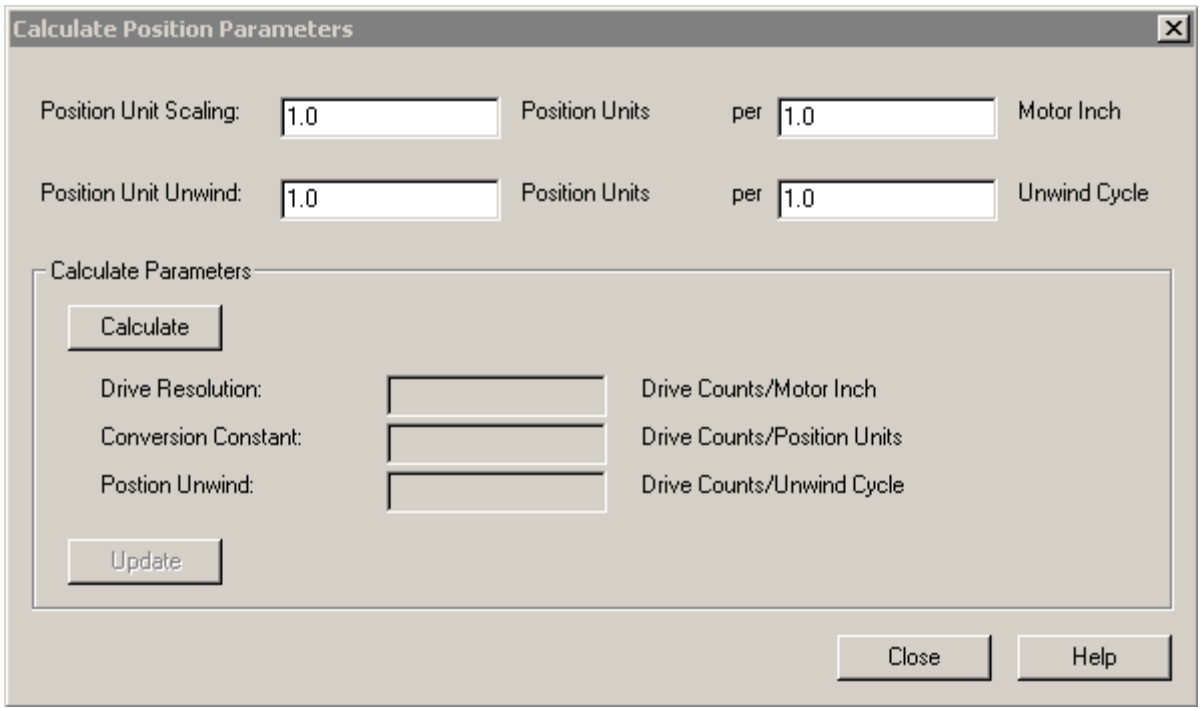
Drive Resolution

Recalculates the resolution based upon the new values entered on this screen.

Conversion Constant

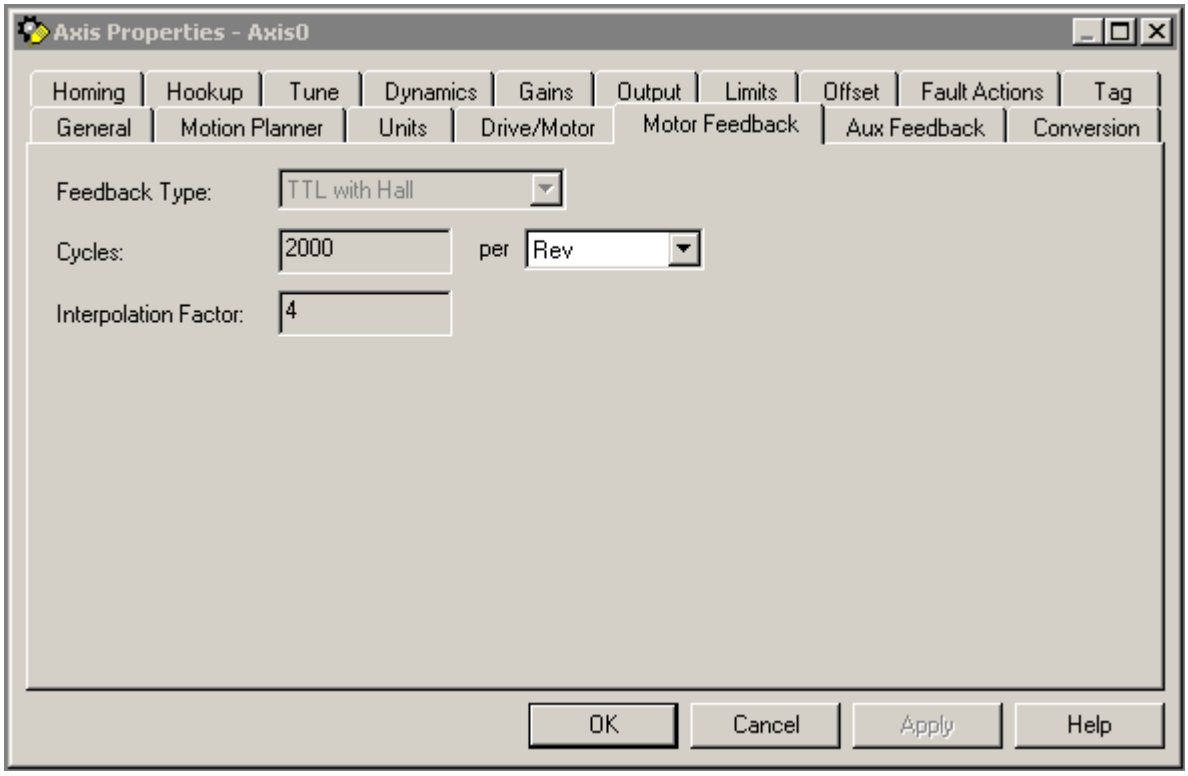
Recalculates the Conversion Constant based upon the new values entered on this screen.

When the Conversion screen has Rotary as the value for Position Mode, clicking on the Calculate button displays the following screen.



Motor Feedback Tab - AXIS_SERVO_DRIVE

Use this tab to configure motor and auxiliary feedback device (if any) parameters, for an axis of the type AXIS_SERVO_DRIVE.



The Axis Configuration selection made on the General tab, and the Loop Configuration selection made on the Drive tab determine which sections of this dialog box – Motor and Auxiliary Feedback – are enabled.

Feedback Type

This field displays the type of feedback associated with the selected motor.

Cycles

The number of cycles of the associated feedback device. This helps the Drive Compute Conversion constant used to convert drive units to feedback counts. Depending on the feedback type you select, this value may be either read-only or editable.

Per

The units used to measure the cycles.

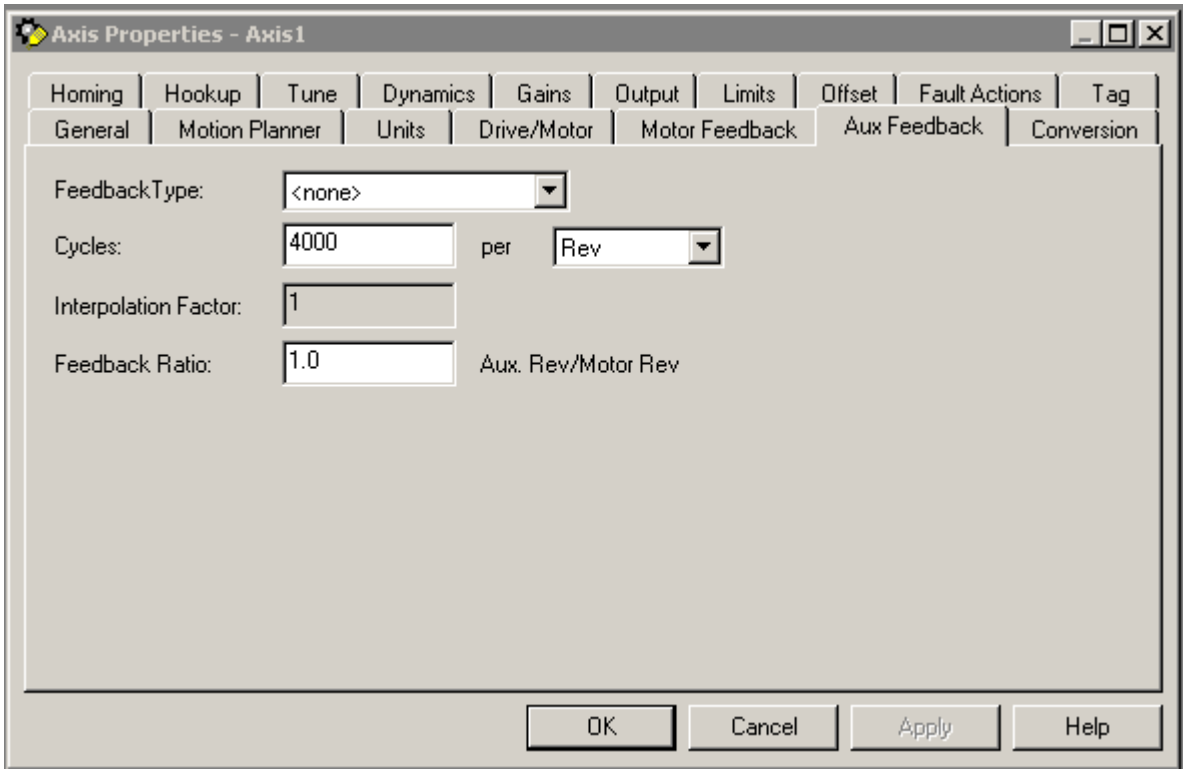
Interpolation Factor

This field displays a fixed, read-only value for each feedback type. This value is used to compute the resolution of the feedback device.

Aux Feedback Tab - AXIS_SERVO_DRIVE

The Auxiliary Feedback tab is enabled only if the Drive tab's Loop Configuration field is set to Aux Feedback Only, Aux Position Servo, Dual Position Servo, Dual Command Servo, or Aux Dual Command Servo.

Use this tab to configure motor and auxiliary feedback device (if any) parameters, for an axis of the type AXIS_SERVO_DRIVE.



Feedback Type

For applications that use auxiliary feedback devices, select the type of auxiliary feedback device type. These are drive dependent.

Cycles

The number of cycles of the auxiliary feedback device. This helps the Drive Compute Conversion constant used to convert drive units to feedback counts. Depending on the feedback type selected, this value may either be read-only or editable.

Per

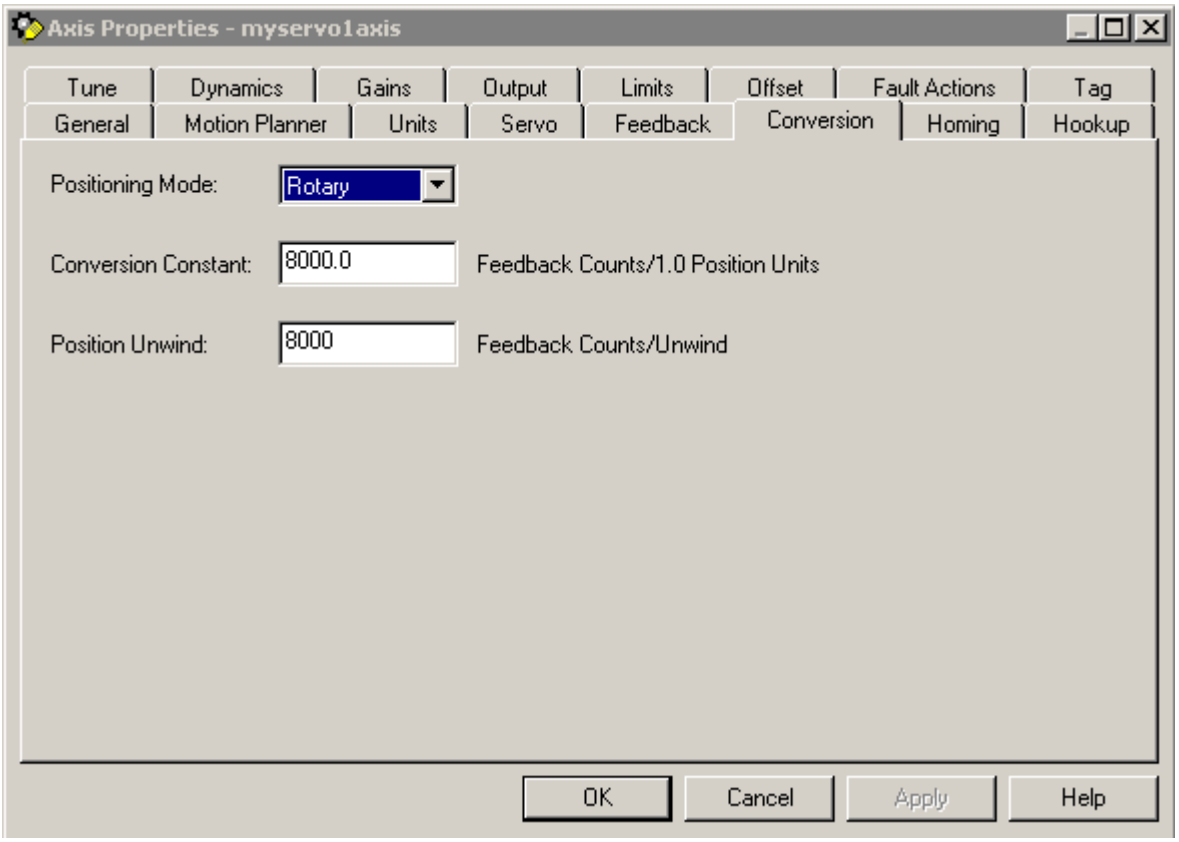
The units used to measure the cycles.

Interpolation Factor

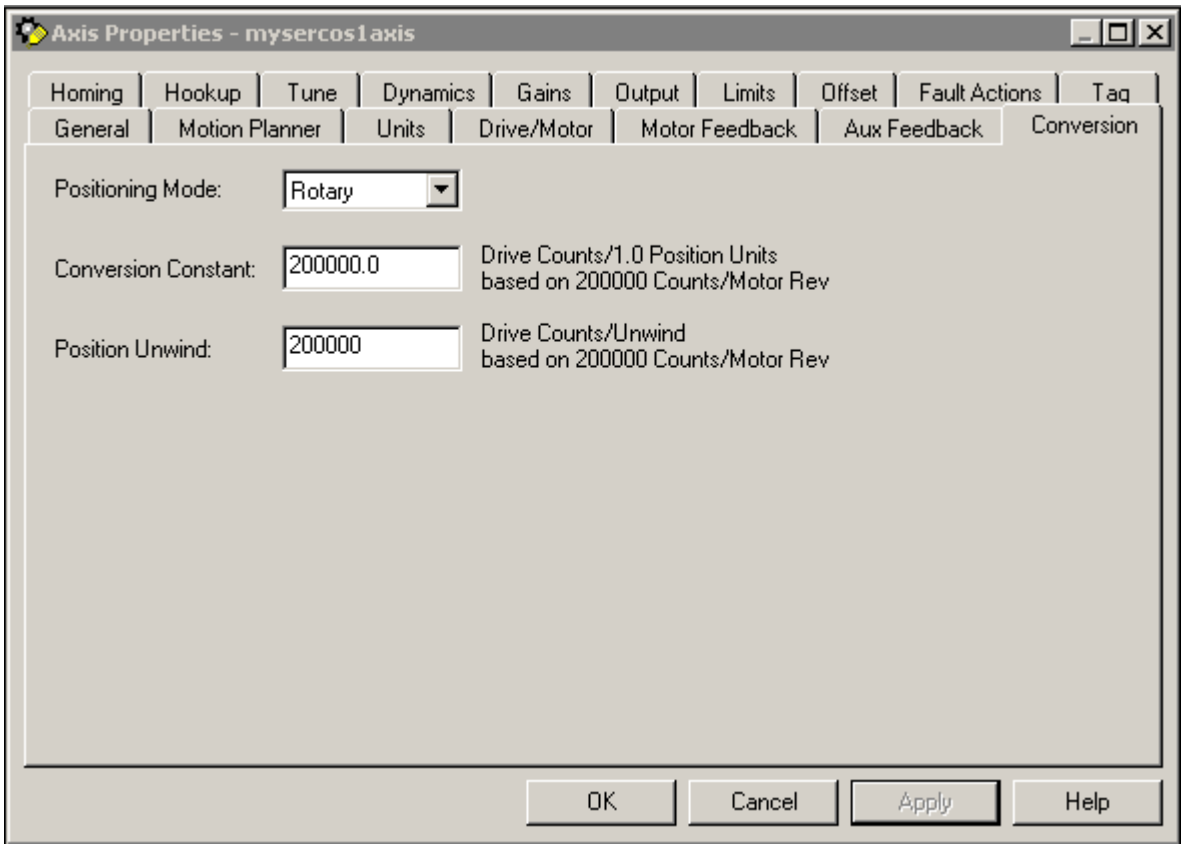
This field displays a fixed constant value for the selected feedback type. This value is used to compute the resolution of the feedback device.

Feedback Ratio Represents the quantitative relationship between the auxiliary feedback device and the motor.

Click on the Conversion tab to access the Axis Properties Conversion dialog.



The differences in the appearance of the Conversion tab screens for the AXIS_SERVO and AXIS_SERVO_DRIVE are the default values for Conversion Constant and Position Unwind and the labels for these values.



Conversion Tab

Use this tab to view/edit the Positioning Mode, Conversion Constant, and if configured as Rotary, the Unwind values for an axis, of the tag types AXIS_SERVO, AXIS_SERVO_DRIVE and AXIS_VIRTUAL.

Positioning Mode

This parameter is not editable for an axis of the data type AXIS_CONSUMED. Instead, this value is set in and taken from a producing axis in a networked Logix processor. This value can be edited for AXIS_SERVO, AXIS_SERVO_DRIVE and AXIS_VIRTUAL.

- Linear - provides a maximum total linear travel of 1 billion feedback counts. With this mode, the unwind feature is disabled and you can limit the linear travel distance traveled by the axis by specifying the positive and negative travel limits for the axis.

- Rotary - enables the rotary unwind capability of the axis. This feature provides infinite position range by unwinding the axis position whenever the axis moves through a complete unwind distance. The number of encoder counts per unwind of the axis is specified by the Position Unwind parameter.

Conversion Constant

Type the number of feedback counts per position unit. This conversion – or "K" – constant allows axis position to be displayed, and motion to be programmed, in the position units set in the Units tab. The conversion constant is used to convert axis position units into feedback counts and vice versa for the AXIS_SERVO type and for the AXIS_SERVO_DRIVE, the number of counts per motor revolution, as set in the Drive Resolution field of the Drive tab.

Position Unwind

This parameter is not editable for an axis of the data type AXIS_CONSUMED. Instead, this value is set in and taken from a producing axis in a networked Logix processor. For a Rotary axis (AXIS_SERVO), this value represents the distance (in feedback counts) used to perform automatic electronic unwind. Electronic unwind allows infinite position range for rotary axes by subtracting the unwind distance from both the actual and command position, every time the axis travels the unwind distance.

For axes of the type AXIS_SERVO_DRIVE:

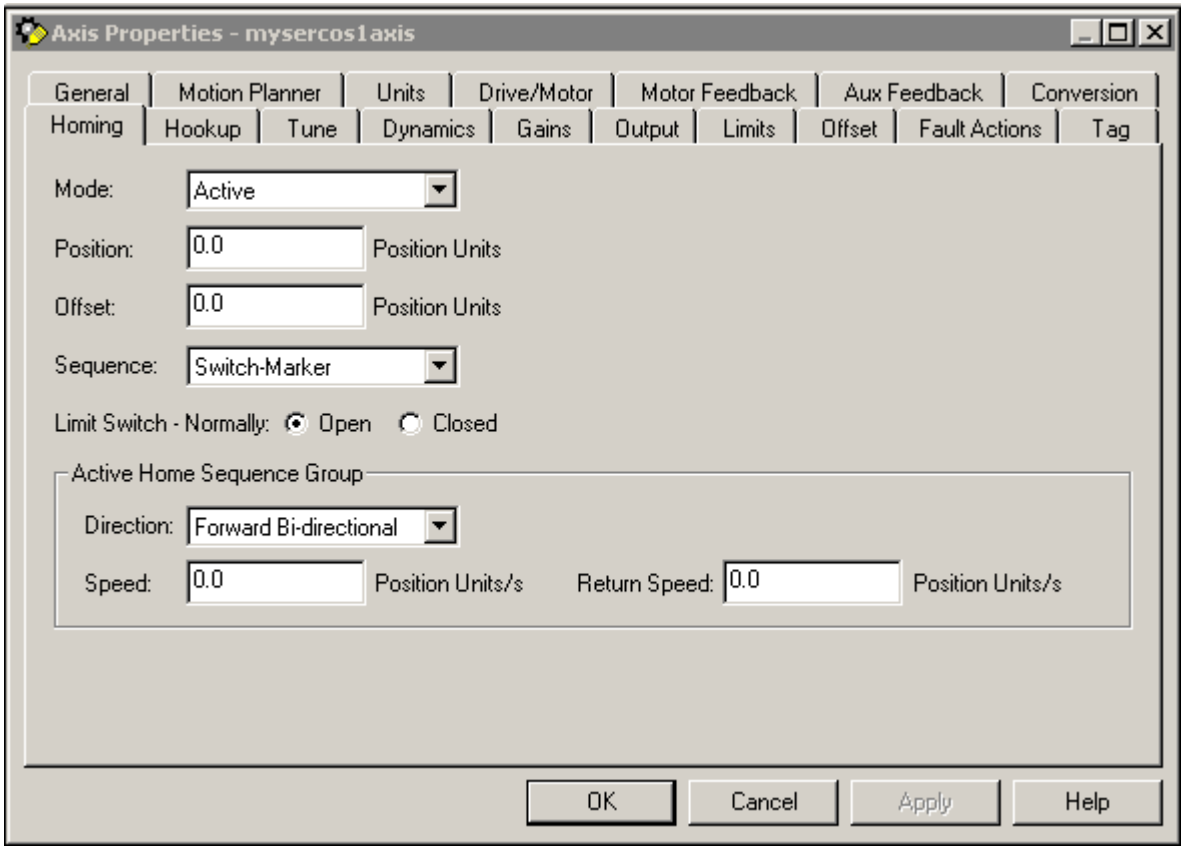
- when you save an edited Conversion Constant or a Drive Resolution value, a message box appears, asking you if you want the controller to automatically recalculate certain attribute settings. (Refer to Conversion Constant and Drive Resolution Attributes.)

- the label indicates the number of counts per motor revolution, as set in the Drive Resolution field of the Drive tab.

Click on Apply to accept your changes.

Homing Tab - AXIS_SERVO

Use this tab to configure the attributes related to homing an axis of the type AXIS_SERVO or AXIS_SERVO_DRIVE.



Mode Select the homing mode.

- Active: In this mode, the desired homing sequence is selected by specifying whether a home limit switch and/or the encoder marker is used for this axis. Active homing sequences always use the trapezoidal velocity profile. For LDT and SSI feedback selections, the only valid Home Sequences for Homing Mode are immediate or switch, as no physical marker exists for the LDT or SSI feedback devices.

- Passive: In this mode, homing redefines the absolute position of the axis on the occurrence of a home switch or encoder marker event. Passive homing is most commonly used to calibrate uncontrolled axes, although it can also be used with controlled axes to create a custom homing sequence. Passive homing, for a given home sequence, works similar to the corresponding active homing sequence, except that no motion is commanded; the controller just waits for the switch and marker events to occur.

- Absolute: (AXIS_SERVO_DRIVE, and AXIS_SERVO when associated with a 1756-HYD02 [LDT feedback] or 1756-M02AS [SSI feedback] module only) In this mode, the absolute homing process establishes the true absolute position of the axis by applying the configured Home Position to the reported position of the absolute feedback device. The only valid Home Sequence for an absolute Homing Mode is immediate. In the LDT and SSI cases, the absolute homing process establishes the true absolute position of the axis by applying the configured Home Position less any enabled Absolute Feedback Offset to the reported position of the absolute feedback device. Prior to execution of the absolute homing process using the MAH instruction, the axis must be in the Axis Ready state with the servo loop disabled.

IMPORTANT

For the SSI feedback transducer no physical marker pulse exists. However, a pseudo marker reference is established by the M02AS module firmware at the feedback device's roll over point. A single-turn Absolute SSI feedback device rolls over at its maximum "turns count" = 1 rev. A multi-turn Absolute SSI feedback device (there are multiple revs or feedback-baseunit-distances) the device rolls over at its maximum "turns count" which is usually either 1024 or 2048.

If you need to establish the rollover of the feedback device, a ladder rung using an SSV to set Home_Sequence equal "Home to marker" with the following parameters: Class Name = SSI_Axis, Attribute_Name = Home_Sequence, and Value = 2 (to Marker) must be added to the application program (cannot be set Axis Properties and must be reset back to its initial value 0 = Immediate or 1 = Switch after establishing the rollover). The Home Sequence = to Marker must be used to allow feedback to travel until the rollover (that is, pseudo marker) is found. This must be done without the motor attached to any axis as this could cause up to Maximum number of turn's before pseudo marker is found.

Position

Type the desired absolute position, in position units, for the axis after the specified homing sequence has been completed. In most cases, this position is set to zero, although any value within the software travel limits can be used. After the homing sequence is complete, the axis is left in this position.

If the Positioning Mode (set in the Conversion tab) of the axis is Linear, then the home position should be within the travel limits, if enabled. If the Positioning Mode is Rotary, then the home position should be less than the unwind distance in position units.

Offset

Type the desired offset (if any) in position units the axis is to move, upon completion of the homing sequence, to reach the home position. In most cases, this value is zero.

Sequence

Select the event that causes the Home Position to be set.

Sequence Type:	Description:
Immediate	Sets the Home Position to the present actual position, without motion.
Switch	Sets the Home Position when axis motion encounters a home limit switch.
Marker	Sets the Home Position when axis encounters an encoder marker.
Switch-Marker	Sets the Home Position when axis first encounters a home limit switch, then encounters an encoder marker.

See the section "Homing Configurations," below, for a detailed description of each combination of homing mode, sequence and direction.

Limit Switch

If a limit switch is used, indicate the normal state of that switch (that is, before being engaged by the axis during the homing sequence):

- Normally Open
- Normally Closed

Direction

For active homing sequences, except for the Immediate Sequence type, select the desired homing direction.

Direction	Description
Forward Uni-directional	The axis jogs in the positive axial direction until a homing event (switch or marker) is encountered, then continues in the same direction until axis motion stops (after decelerating or moving the Offset distance).

Direction	Description
Forward Bi-directional	The axis jogs in the positive axial direction until a homing event (switch or marker) is encountered, then reverses direction until motion stops (after decelerating or moving the Offset distance).
Reverse Uni-directional	The axis jogs in the negative axial direction until a homing event (switch or marker) is encountered, then continues in the same direction until axis motion stops (after decelerating or moving the Offset distance).
Reverse Bi-directional	The axis jogs in the negative axial direction until a homing event (switch or marker) is encountered, then reverses direction until motion stops (after decelerating or moving the Offset distance).

Speed

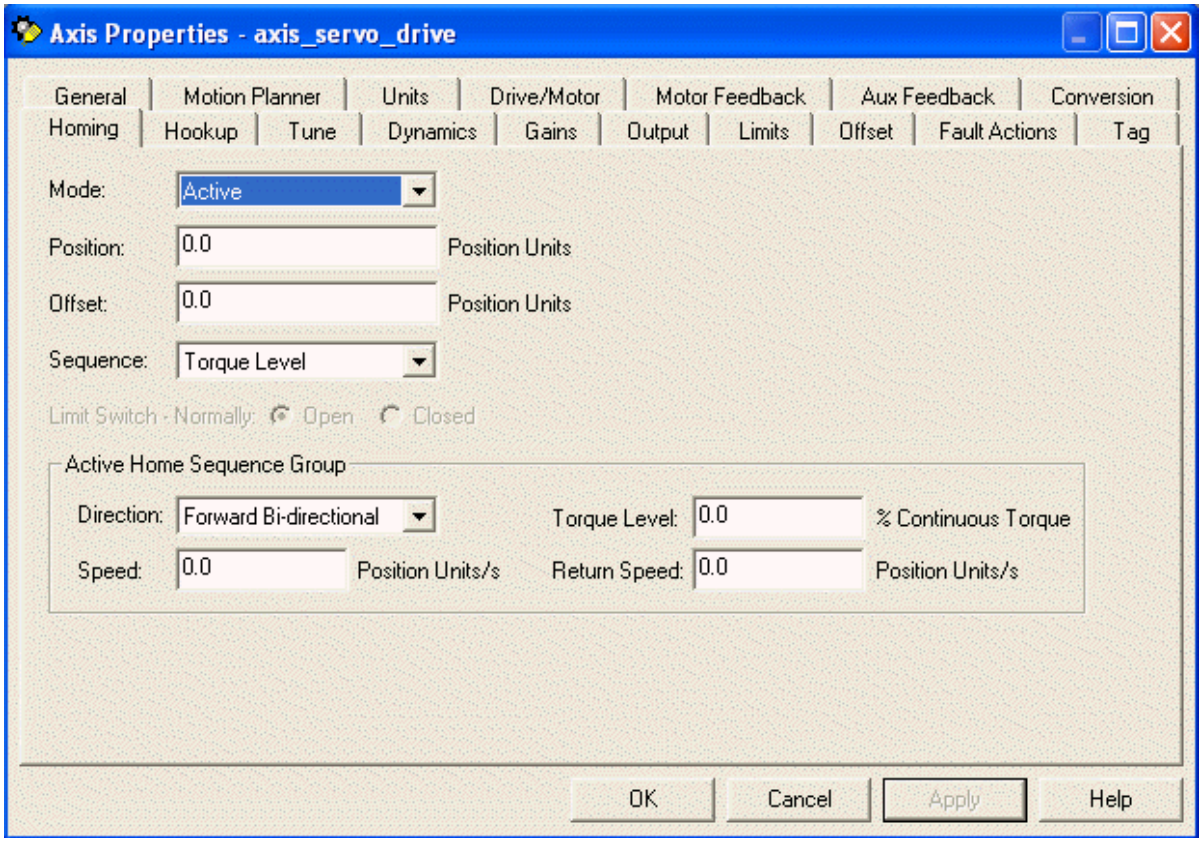
Type the speed of the jog profile used in the first leg of an active homing sequence. The homing speed specified should be less than the maximum speed and greater than zero.

Return Speed

The speed of the jog profile used in the return leg(s) of an active homing sequence. The home return speed specified should be less than the maximum speed and greater than zero.

Homing Tab - AXIS_SERVO_DRIVE

Use this tab to configure the attributes related to homing an axis of the type AXIS_SERVO_DRIVE.



Mode Select the homing mode.

- Active: In this mode, the desired homing sequence is selected by specifying whether a home limit switch and/or the encoder marker is used for this axis. Active homing sequences always use the trapezoidal velocity profile. For LDT and SSI feedback selections, the only valid Home Sequences for Homing Mode are immediate or switch, as no physical marker exists for the LDT or SSI feedback devices.

- Passive: In this mode, homing redefines the absolute position of the axis on the occurrence of a home switch or encoder marker event. Passive homing is most commonly used to calibrate uncontrolled axes, although it can also be used with controlled axes to create a custom homing sequence. Passive homing, for a given home sequence, works similar to the corresponding active homing sequence, except that no motion is commanded; the controller just waits for the switch and marker events to occur.

- Absolute: (AXIS_SERVO_DRIVE, and AXIS_SERVO when associated with a 1756-HYD02 [LDT feedback] or 1756-M02AS [SSI feedback] module only) In this mode, the absolute homing process establishes the true absolute position of the axis by applying the configured Home Position to the reported position of the absolute feedback device. The only valid Home Sequence for an absolute Homing Mode is immediate. In the LDT and SSI cases, the absolute homing process establishes the true absolute position of the axis by applying the configured Home Position less any enabled Absolute Feedback Offset to the reported position of the absolute feedback device. Prior to execution of the absolute homing process using the MAH instruction, the axis must be in the Axis Ready state with the servo loop disabled.

IMPORTANT

For the SSI feedback transducer no physical marker pulse exists. However, a pseudo marker reference is established by the M02AS module firmware at the feedback device's roll over point. A single-turn Absolute SSI feedback device rolls over at its maximum "turns count" = 1 rev. A multi-turn Absolute SSI feedback device (there are multiple revs or feedback-baseunit-distances) the device rolls over at its maximum "turns count" which is usually either 1024 or 2048.

If you need to establish the rollover of the feedback device, a ladder rung using an SSV to set Home_Sequence equal "Home to marker" with the following parameters: Class Name = SSI_Axis, Attribute_Name = Home_Sequence, and Value = 2 (to Marker) must be added to the application program (cannot be set Axis Properties and must be reset back to its initial value 0 = Immediate or 1 = Switch after establishing the rollover). The Home Sequence = to Marker must be used to allow feedback to travel until the rollover (that is, pseudo marker) is found. This must be done without the motor attached to any axis as this could cause up to Maximum number of turn's before pseudo marker is found.

Position

Type the desired absolute position, in position units, for the axis after the specified homing sequence has been completed. In most cases, this position is set to zero, although any value within the software travel limits can be used. After the homing sequence is complete, the axis is left in this position.

If the Positioning mode (set in the Conversion tab) of the axis is Linear, then the home position should be within the travel limits, if enabled. If the Positioning mode is Rotary, then the home position should be less than the unwind distance in position units.

Offset Type the desired offset (if any) in position units the axis is to move, upon completion of the homing sequence, to reach the home position. In most cases, this value is zero.

Sequence Select the event that causes the Home Position to be set.

Sequence Type:	Description:
Immediate	Sets the Home Position to the present actual position, without motion.
Switch	Sets the Home Position when axis motion encounters a home limit switch.
Marker	Sets the Home Position when axis encounters an encoder marker.
Switch-Marker	Sets the Home Position when axis first encounters a home limit switch, then encounters an encoder marker.
Torque Level	Sets the Home Position when the specified Homing Torque level is achieved on the assigned axis.
Torque Level-marker	Sets the Home Position when the specified Homing Torque level is achieved on the assigned axis, only after the axis encounters an encoder marker.

Refer to the section Homing Configurations, for a detailed description of each combination of homing mode, sequence and direction.

Limit Switch If a limit switch is used, indicate the normal state of that switch (that is, before being engaged by the axis during the homing sequence).

- Normally Open
- Normally Closed

Direction

For active homing sequences, except for the Immediate Sequence type, select the desired homing direction.

Direction	Description
Forward Uni-directional	The axis jogs in the positive axial direction until a homing event (switch or marker) is encountered, then continues in the same direction until axis motion stops (after decelerating or moving the Offset distance).
Forward Bi-directional	The axis jogs in the positive axial direction until a homing event (switch or marker) is encountered, then reverses direction until motion stops (after decelerating or moving the Offset distance).
Reverse Uni-directional	The axis jogs in the negative axial direction until a homing event (switch or marker) is encountered, then continues in the same direction until axis motion stops (after decelerating or moving the Offset distance).
Reverse Bi-directional	The axis jogs in the negative axial direction until a homing event (switch or marker) is encountered, then reverses direction until motion stops (after decelerating or moving the Offset distance).

Speed

Type the speed of the jog profile used in the first leg of an active homing sequence. The homing speed specified should be less than the maximum speed and greater than zero.

Torque Level

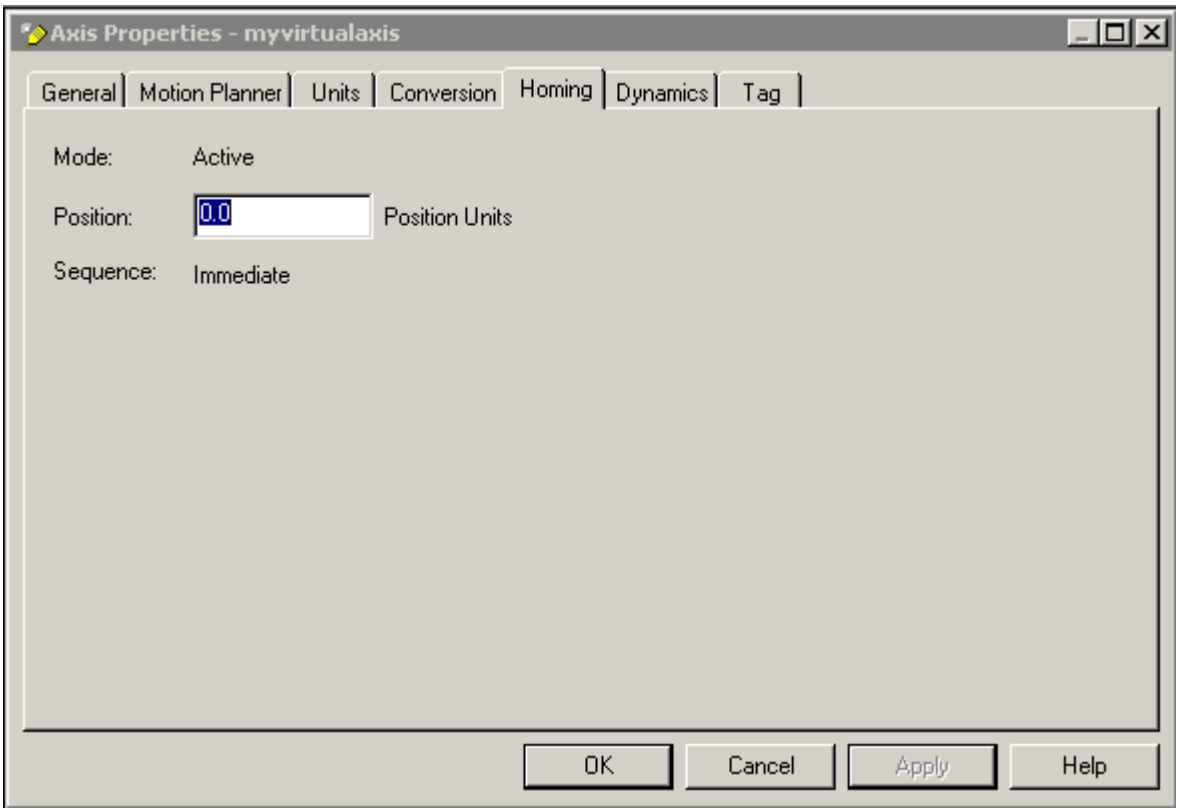
The torque level, with units % continuous torque, that the axis motor must reach to complete the Home-to-Torque sequence. This feature is only available on the Kinetix family of drives.

Return Speed

The speed of the jog profile used in the return leg(s) of an active homing sequence. The home return speed specified should be less than the maximum speed and greater than zero.

Homing Tab - AXIS_VIRTUAL

Use this tab to configure the attributes related to homing an axis of the type AXIS_VIRTUAL.



Only an Active Immediate Homing sequence can be performed for an axis of the type AXIS_VIRTUAL. When this sequence is performed, the controller immediately enables the servo drive and assigns the Home Position to the current axis actual position and command position. This homing sequence produces no axis motion.

Mode

This read-only parameter is always set to Active.

Position

Type the desired absolute position, in position units, for the axis after the specified homing sequence has been completed. In most cases, this position is set to zero, although any value within the software travel limits can be used. After the homing sequence is complete, the axis is left at this position.

If the Positioning Mode (set in the Conversion tab) of the axis is Linear, then the home position should be within the travel limits, if enabled. If the Positioning Mode is Rotary, then the home position should be less than the unwind distance in position units.

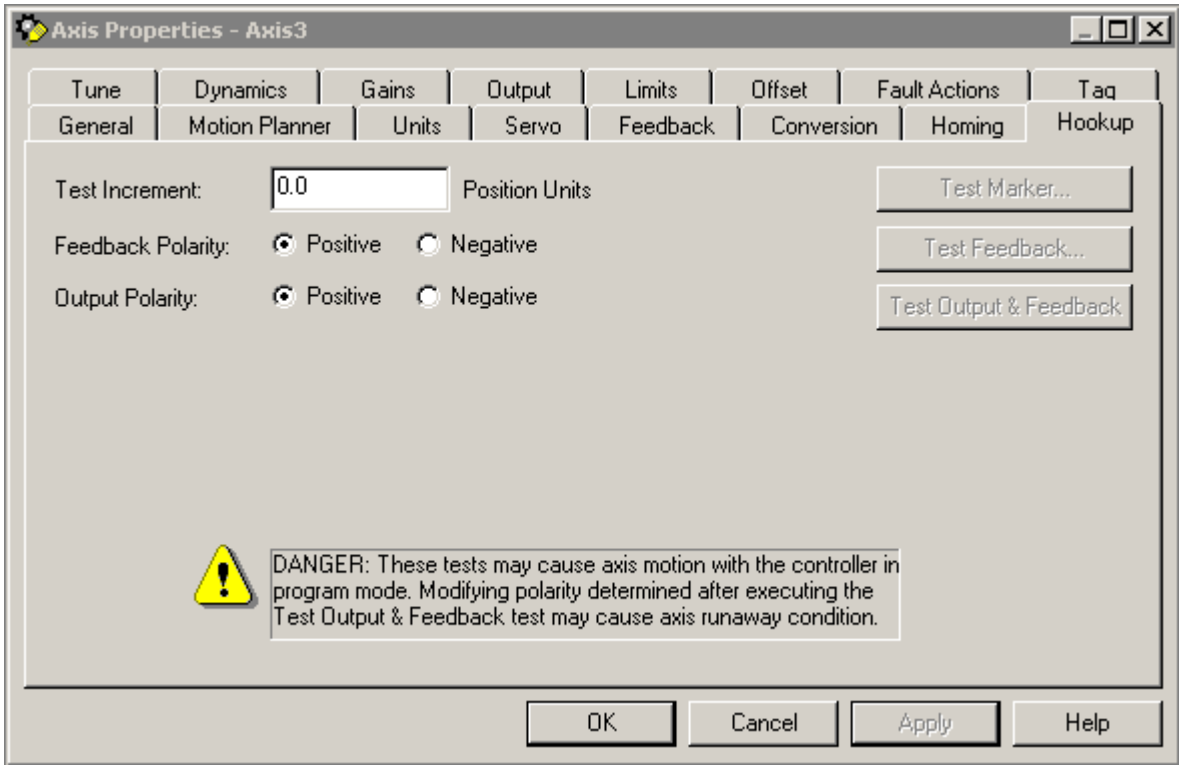
Sequence

This read-only parameter is always set to Immediate.

Hookup Tab - AXIS_SERVO

Use this tab to configure and initiate axis hookup and marker test sequences for an axis of the type AXIS_SERVO.

When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.



Test Increment

Specifies the amount of distance traversed by the axis when executing the Output & Feedback test. The default value is set to approximately a quarter of a revolution of the motor in position units.

Feedback Polarity

The polarity of the encoder feedback, this field is automatically set by executing either the Feedback Test or the Output & Feedback Test.

- Positive
- Negative

When properly configured, this setting insures that axis Actual Position value increases when the axis is moved in the user defined positive direction. This bit can be configured automatically using the MRHD and MAHD motion instructions.

ATTENTION

Modifying automatically input polarity values by running the Feedback or Output & Feedback Tests can cause a runaway condition resulting in unexpected motion, damage to the equipment, and physical injury or death.

Output Polarity

The polarity of the servo output to the drive, this field is automatically set by executing the Output & Feedback Test.

- Positive
- Negative

When properly configured, this setting and the Feedback Polarity setting insure that, when the axis servo loop is closed, it is closed as a negative feedback system and not an unstable positive feedback system. This bit can be configured automatically using the MRHD and MAHD motion instructions.

Test Marker

Runs the Marker test, which ensures that the encoder A, B, and Z channels are connected correctly and phased properly for marker detection. When the test is initiated, you must manually move the axis one revolution for the system to detect the marker. If the marker is not detected, check the encoder wiring and try again.

Test Feedback

Runs the Feedback Test, which checks and, if necessary, reconfigures the Feedback Polarity setting. When the test is initiated, you must manually move the axis one revolution for the system to detect the marker. If the marker is not detected, check the encoder wiring and try again.

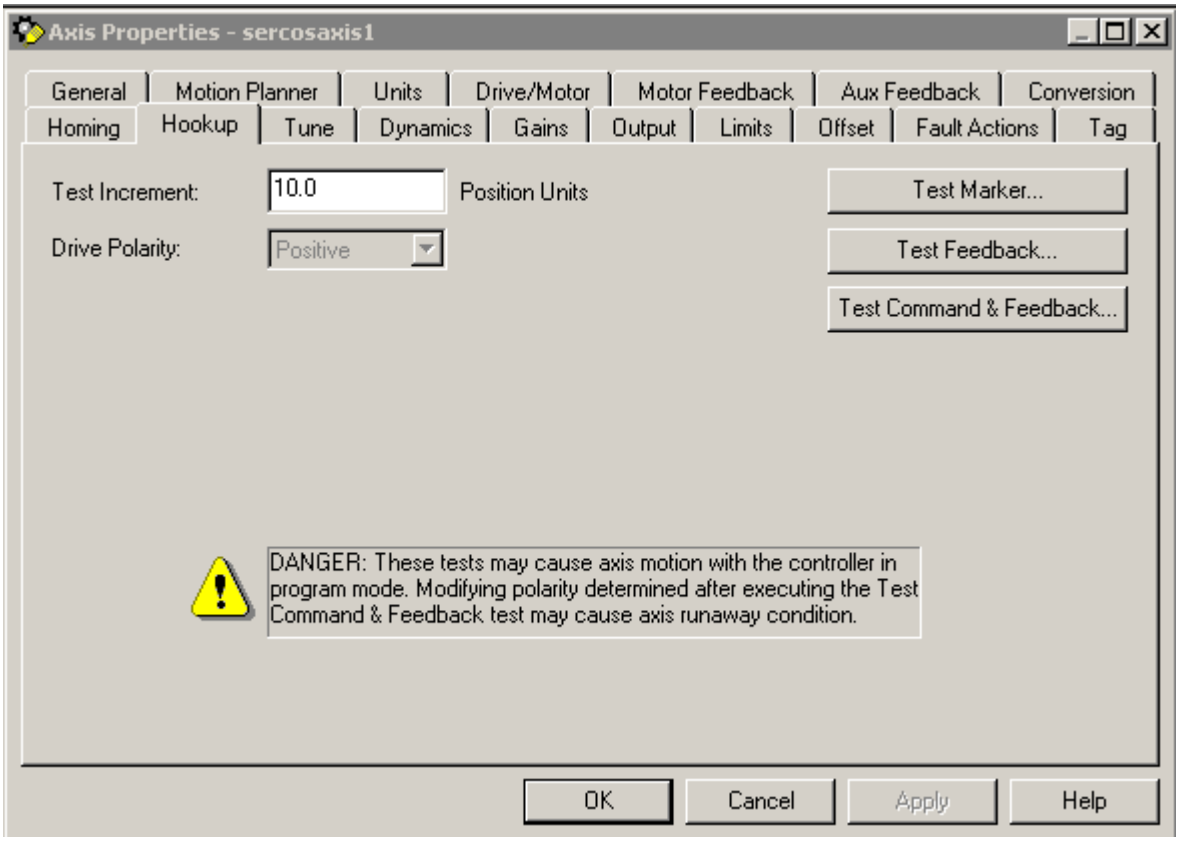
Test Output & Feedback

Runs the Output & Feedback Test, which checks and, if necessary, reconfigures both the polarity of encoder feedback (the Feedback Polarity setting) and the polarity of the servo output to the drive (the Output Polarity setting), for an axis configured for Servo operation in the General tab.

Executing any test operation automatically saves all changes to axis properties.

Hookup Tab Overview - AXIS_SERVO_DRIVE

Use this tab to configure and initiate axis hookup and marker test sequences for an axis of the type AXIS_SERVO_DRIVE.



When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

Test Increment

Specifies the amount of distance traversed by the axis when executing the Command & Feedback test. The default value is set to approximately a quarter of a revolution of the motor in position units.

Drive Polarity

The polarity of the servo loop of the drive, set by executing the Command & Feedback Test.

- Positive
- Negative

Proper wiring guarantees that the servo loop is closed with negative feedback. However there is no guarantee that the servo drive has the same sense of forward direction as the user for a given application. Negative Polarity inverts

the polarity of both the command position and actual position data of the servo drive. Thus, selecting either Positive or Negative Drive Polarity makes it possible to configure the positive direction sense of the drive to agree with that of the user. This attribute can be configured automatically using the MRHD and MAHD motion instructions.

ATTENTION

Modifying polarity values, automatically input by running the Command & Feedback Test, can cause a runaway condition.

Test Marker

Runs the Marker test, which ensures that the encoder A, B, and Z channels are connected correctly and phased properly for marker detection. When the test is initiated, you must manually move the axis one revolution for the system to detect the marker. If the marker is not detected, check the encoder wiring and try again.

Test Feedback

Runs the Feedback Test, which checks and, if necessary, reconfigures the Feedback Polarity setting. When the test is initiated, you must manually move the axis one revolution for the system to detect the marker. If the marker is not detected, check the encoder wiring and try again.

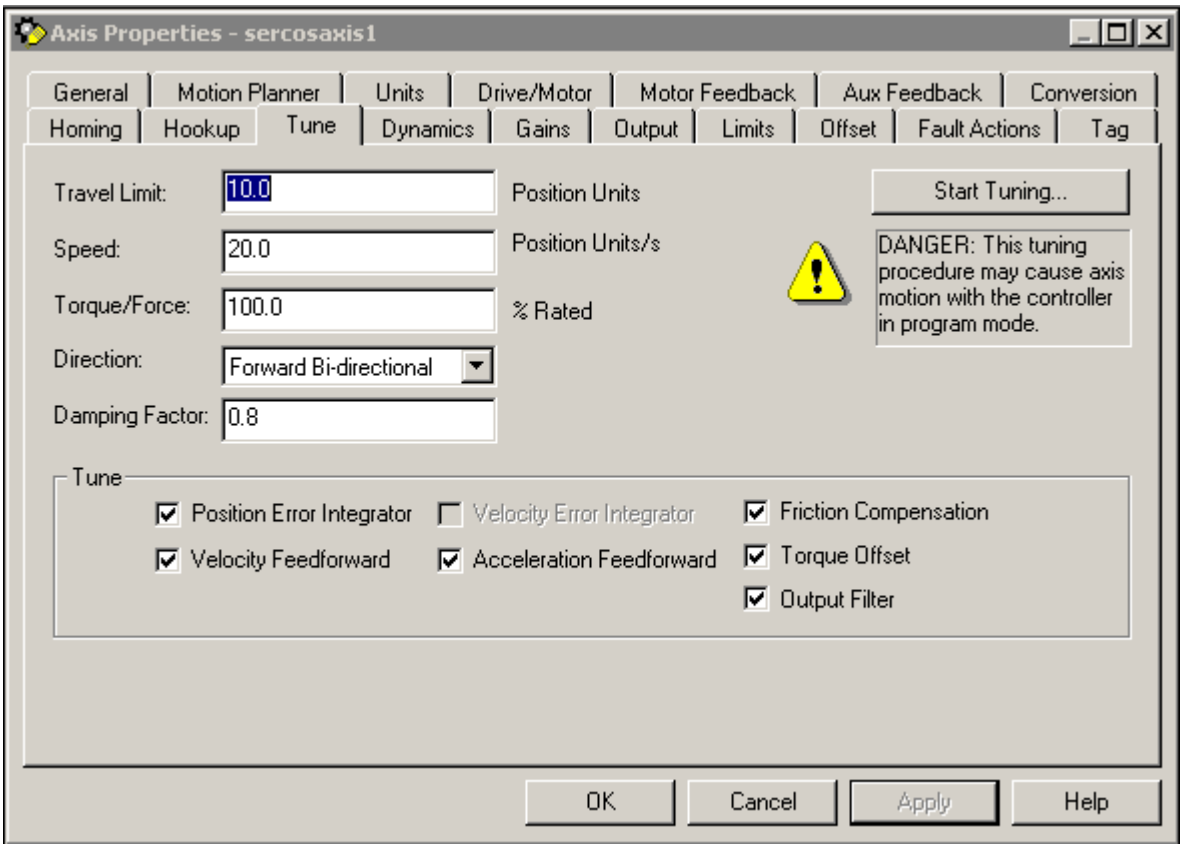
Test Command & Feedback

Runs the Command & Feedback Test, which checks and, if necessary, reconfigures both the polarity of encoder feedback (the Feedback Polarity setting) and the polarity of the servo output to the drive (the Output Polarity setting), for an axis configured for Servo operation in the General tab.

Executing any test operation automatically saves all changes to axis properties.

Tune Tab - AXIS_SERVO, AXIS_SERVO_DRIVE

Use this tab to configure and initiate the axis tuning sequence for an axis of the types AXIS_SERVO or AXIS_SERVO_DRIVE.



Travel Limit

Specifies a limit to the excursion of the axis during the tune test. If the servo module determines that the axis is not able to complete the tuning process before exceeding the tuning travel limit, it terminates the tuning profile and report that this limit was exceeded.

Speed

Determines the maximum speed for the tune process. This value should be set to the desired maximum operating speed of the motor (in engineering units) prior to running the tune test.

Torque/Force (AXIS_SERVO_DRIVE)

The maximum torque of the tune test. Force is used only when a linear motor is connected to the application. This attribute should be set to the desired maximum safe torque level prior to running the tune test. The default value is 100%, which yields the most accurate measure of the acceleration and deceleration capabilities of the system.

In some cases a lower tuning torque limit value may be desirable to limit the stress on the mechanics during the tuning procedure. In this case the acceleration and deceleration capabilities of the system are extrapolated based on the ratio of the tuning torque to the maximum torque output of the system. Extrapolation error increases as the Tuning Torque value decreases.

Torque (AXIS_SERVO)

The maximum torque of the tune test. This attribute should be set to the desired maximum safe torque level prior to running the tune test. The default value is 100%, which yields the most accurate measure of the acceleration and deceleration capabilities of the system.

In some cases a lower tuning torque limit value may be desirable to limit the stress on the mechanics during the tuning procedure. In this case the acceleration and deceleration capabilities of the system are extrapolated based on the ratio of the tuning torque to the maximum torque output of the system. Extrapolation error increases as the Tuning Torque value decreases.

Direction

The direction of the tuning motion profile.

- Forward Uni-directional – the tuning motion profile is initiated in the forward tuning direction only.
- Forward Bi-directional – the tuning motion profile is first initiated in the forward tuning direction and then, if successful, is repeated in the reverse direction. Information returned by the Bi-directional Tuning profile can be used to tune Friction Compensation and Torque Offset.
- Reverse Uni-directional – the tuning motion profile is initiated in the reverse tuning direction only.
- Reverse Bi-directional – the tuning motion profile is first initiated in the reverse tuning direction and then, if successful, is repeated in the forward direction. Information returned by the Bi-directional Tuning profile can be used to tune Friction Compensation and Torque Offset.

Damping Factor

Specifies the dynamic response of the servo axis. The default is set to 0.8. When gains are tuned using a small damping factor, a step response test performed on the axis may generate uncontrolled oscillation. The gains generated using a larger damping factor would produce a system step response that has no overshoot and is stable, but may be sluggish in response to changes.

The tuning procedure uses the Damping Factor that is set in this field. However, when the controller recalculates certain attributes in response to a Motor Catalog Number change (on the Motor/Feedback tab), the controller uses the default Damping Factor value of 0.8, and not a different value set in this field.

Tune Select the gains to be determined by the tuning test.

- Position Error Integrator – determines whether or not to calculate a value for the Position Integral Gain.
- Velocity Feedforward – determines whether or not to calculate a value for the Velocity Feedforward Gain.
- Velocity Error Integrator – determines whether or not to calculate a value for the Velocity Integral Gain.
- Acceleration Feedforward – determines whether or not to calculate a value for the Acceleration Feedforward Gain.
- Friction Compensation – determines whether or not to calculate a value for the Friction Compensation Gain.
- Torque Offset – determines whether or not to calculate a value for the Torque Offset. This tuning configuration is only valid if configured for bidirectional tuning.
- Output Filter – determines whether or not to calculate a value for the Output Filter Bandwidth.

Start Tuning Click on this button to begin the tuning test. If the tuning process completes successfully the following attributes are set.

On this tab	These attributes are set
Gains tab	Velocity Feedforward Gain (if checked under Tune, above) Acceleration Feedforward Gain (if checked under Tune, above) Position Proportional Gain Position Integral Gain (if checked under Tune, above) Velocity Proportional Gain Velocity Integral Gain (if checked under Tune, above)
Dynamics tab	Maximum Speed Maximum Acceleration Maximum Deceleration Maximum Acceleration Jerk Maximum Deceleration Jerk
Output tab	Torque Scaling Velocity Scaling (AXIS_SERVO only) Low Pass Output Filter
Limits	Position Error Tolerance

The Tune Bandwidth dialog opens for Servo drives, where you can tweak bandwidth values.

During tuning, if the controller detects a high degree of tuning inertia, it enables the Low Pass Output Filter and calculates and sets a value for Low Pass Output Filter Bandwidth.

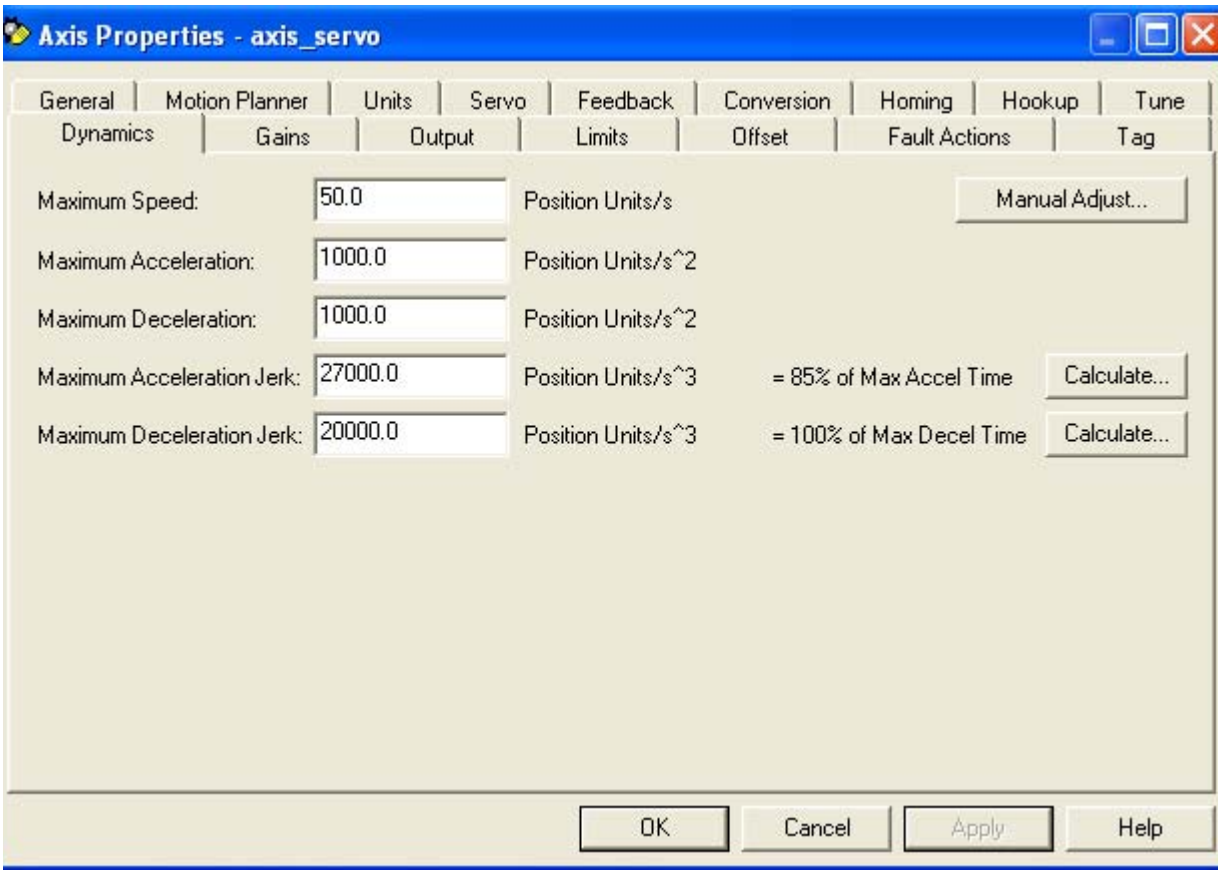
Executing a Tune operation automatically saves all changes to axis properties.

ATTENTION

This tuning procedure may cause axis motion with the controller in program mode. Unexpected motion may cause damage to the equipment, personal injury, or death.

Dynamics Tab - AXIS_SERVO, AXIS_SERVO_DRIVE, AXIS_VIRTUAL

Use this tab to view or edit the dynamics related parameters for an axis of the type AXIS_SERVO or AXIS_SERVO_DRIVE configured for Servo operations in the General tab of this dialog box, or AXIS_VIRTUAL.



IMPORTANT The parameters on this tab can be edited in either of two ways:

- edit on this tab by typing your parameter changes and then clicking on OK or Apply to save your edits.
- edit in the Manual Adjust dialog: click on the Manual Adjust button to open the Manual Adjust dialog to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When RSLogix 5000 software is offline, the following parameters can be edited and the program saved to disk using either the Save command or by clicking on the Apply button. You must re-download the edited program to the controller before it can be run.

Maximum Speed

The steady-state speed of the axis, it is initially set to Tuning Speed by the tuning process. This value is typically set to about 90% of the maximum speed rating of the motor. This provides sufficient "head-room" for the axis to operate at all times within the speed limitations of the motor. The Maximum Speed value entered is used when the motion instruction is set with Speed Units=% of Maximum. If a motion instruction has a Speed Units= units per sec value entered then, the speed is taken from the motion instruction faceplate.

Maximum Acceleration

The maximum acceleration rate of the axis, in Position Units/second, it is initially set to about 85% of the measured tuning acceleration rate by the tuning process. If set manually, this value should typically be set to about 85% of the maximum acceleration rate of the axis. This provides sufficient "head-room" for the axis to operate at all times within the acceleration limits of the drive and motor. The Maximum Acceleration value entered is used when the motion instruction is set with Accel Units=% of Maximum. When a motion instruction is configured with Accel Units=units per sec^2 field then, the Maximum Acceleration is taken from the motion instruction faceplate.

Maximum Deceleration

The maximum deceleration rate of the axis, in Position Units/second, it is initially set to approximately 85% of the measured tuning deceleration rate by

the tuning process. If set manually, this value should typically be set to about 85% of the maximum deceleration rate of the axis. This provides sufficient "head-room" for the axis to operate at all times within the deceleration limits of the drive and motor. The Maximum Deceleration value entered is used when the motion instruction is set with decel Units=% of Maximum. When a motion instruction is configured with Decel Units=units per sec^2 field then, the Maximum Deceleration is taken from the motion instruction faceplate.

Maximum Acceleration Jerk

The jerk parameters only apply to S-curve profile moves using these instructions:

- MAJ
- MAM
- MAS
- MCD

The Maximum Acceleration Jerk rate of the axis, in Position Units/$second^3$, defaults to 100% of the maximum acceleration time after tuning. The speed and acceleration rate for this calculation are determined during S-curve the tuning process.

$$\frac{MaxAccel^2}{Speed} = \text{Maximum Acceleration Jerk}$$

The Maximum Accel Jerk value entered is used when the motion instruction is set with Jerk Units=% of Maximum. When a Single-axis Motion Instruction has Jerk Units=units per sec^3 then the maximum acceleration jerk value is derived from the motion instruction faceplate. The jerk units for the motion instruction also allow for Jerk Units=% of Time, with 100% of Time. This means that the entire S-curve move will have Jerk limiting. This is the default mode. An S-curve move with 0% of Time will result in a trapezoidal profile, and have 0% Jerk limiting. If set manually, enter the value in units=Position Units/$second^3$ units. You can also use the Calculate button to view this value in terms of units=% of Time.

Maximum Deceleration Jerk

The jerk parameters only apply to S-curve profile moves using these instructions:

- MAJ
- MAM
- MAS
- MCD

The Maximum Deceleration Jerk rate of the axis, in Position Units/second^3, defaults to 100% of the maximum deceleration time after tuning. The speed and deceleration rate for the calculation are determined during the tuning process.

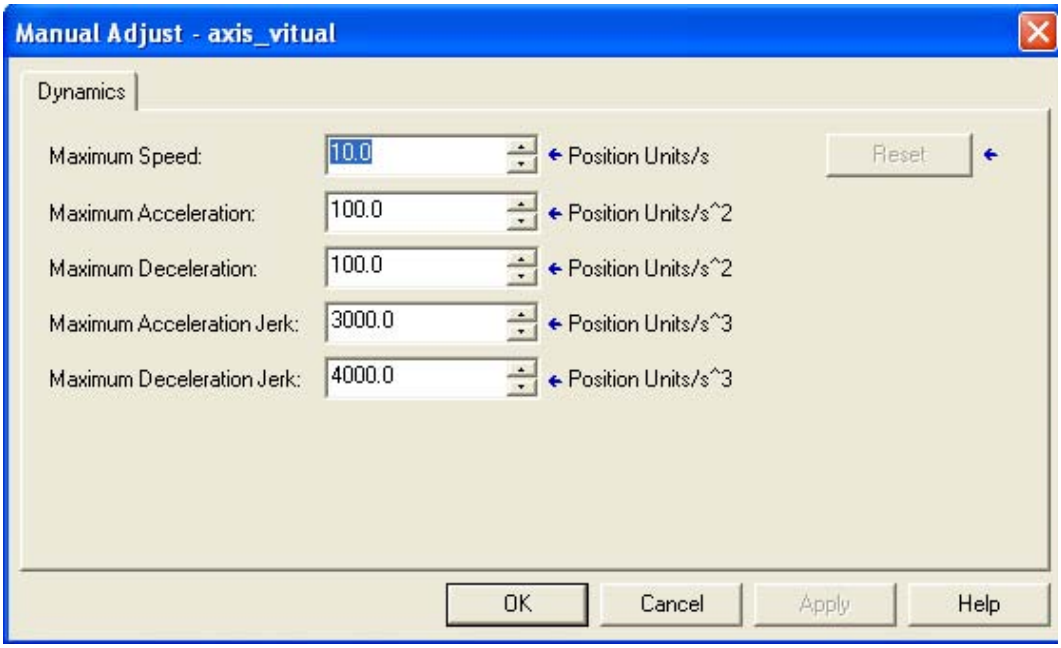
$$\frac{\text{MaxDecel}^2}{\text{Speed}} = \text{Maximum Deceleration Jerk}$$

The Maximum Decel Jerk value entered is used when the motion instruction is set with Jerk Units=% of Maximum. When a Single-axis motion instruction has Jerk Units=units per sec^3 then the Max Deceleration Jerk value is derived from the Motion Instruction faceplate. The jerk units for the motion instruction also allow for Jerk Units=% of Time, with 100% of Time meaning the entire S-curve move will have Jerk limiting, which is the default mode. An S-curve move with 0% of Time will result in a trapezoidal profile, and have 0% Jerk limiting. If set manually, enter the value in units=Position Units/second^3 units. You can also use the optional Calculate button to view the value in terms of units=% of Time.

Manual Adjust

Click on this button to open the Dynamics tab of the Manual Adjust dialog for online editing of the Maximum Speed, Maximum Acceleration, Maximum Deceleration, Maximum Acceleration Jerk, and Maximum Deceleration Jerk. When values are changed on this dialog, either by manually changing the spin

control or entering numeric values, the new values are instantaneously sent to the controller.



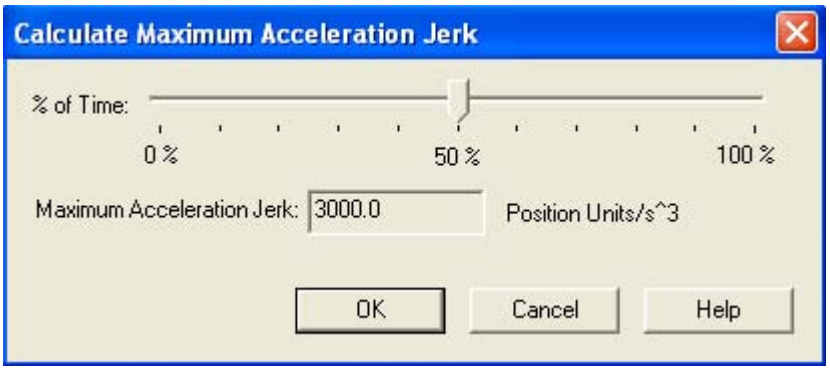
The Manual Adjust button is disabled when RSLogix 5000 software is in Wizard mode, and when offline edits to the above parameters have not yet been saved or applied.

Calculate Button

This dialog lets you set and view the Maximum Acceleration or Deceleration Jerk in Jerk Units=% of Time. Use the slider to select the value unit=% of Time. The numeric value in the Maximum Accel\Decel Jerk status box updates as the slider is moved. Click on the OK button to accept the new value, or click the Cancel button to leave without changing the value.

The Unit=% of Time is allowed for Jerk limiting only via the Instruction Faceplate. Only the Profile=S-curve allows Jerk control (Programmable

S-curve). The units for programming Jerk limiting are more easily expressed in terms of % of Time rather than Position Units/s^3.

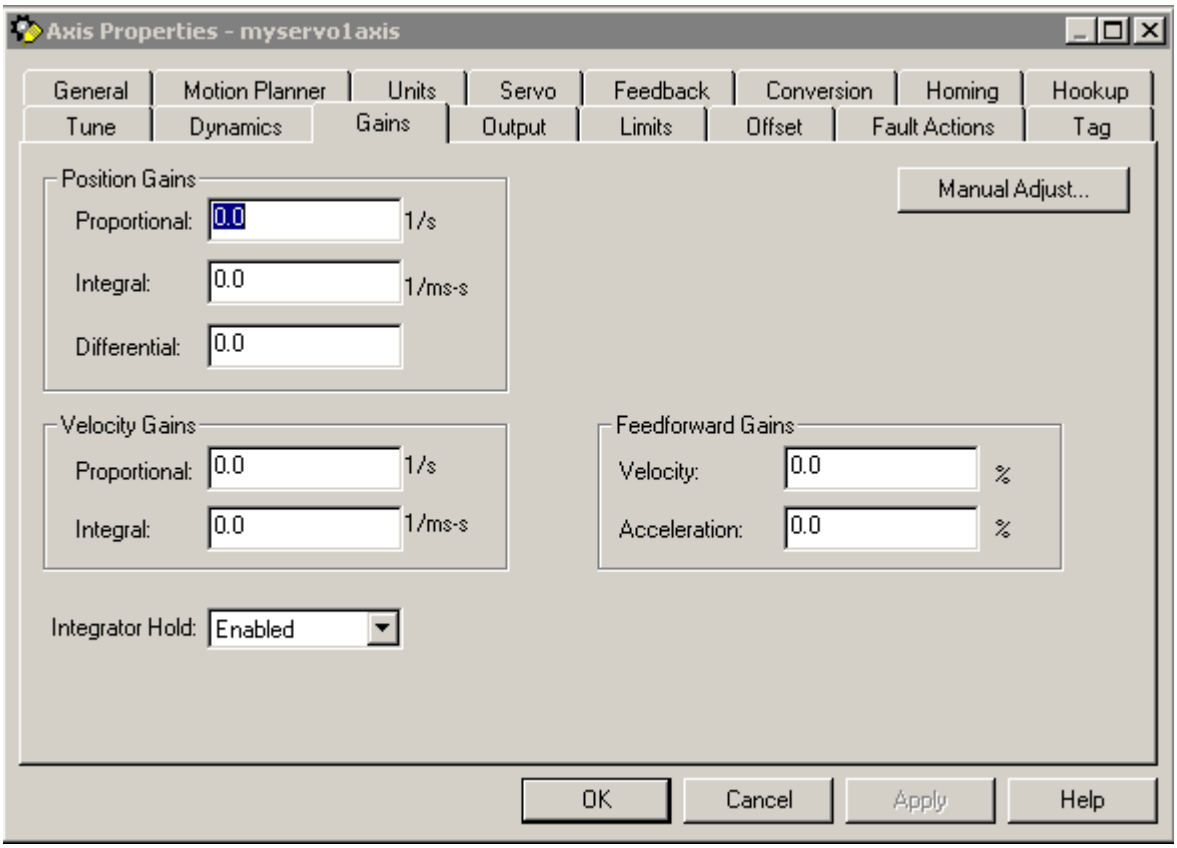


Gains Tab - AXIS_SERVO

Use this tab to perform these offline functions.

- adjust, or "tweak" gain values that have been automatically set by the tuning process (in the Tune tab of this dialog)
- manually configure gains for the velocity and position loops

for an axis of the type AXIS_SERVO, which has been configured for Servo operations (set in the General tab of this dialog box), with Position Loop Configuration.



The drive module uses a nested digital servo control loop consisting of a position loop with proportional, integral and feed-forward gains around an optional digitally synthesized inner velocity loop. The parameters on this tab can be edited in either of two ways:

- edit on this tab by typing your parameter changes and then clicking on OK or Apply to save your edits
- edit in the Manual Adjust dialog: click on the Manual Adjust button to open the Manual Adjust dialog to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When RSLogix 5000 software is offline, the following parameters can be edited and the program saved to disk using either the Save command or by clicking on the Apply button. You must re-download the edited program to the controller before it can be run.

Proportional (Position) Gain

Position Error is multiplied by the Position Loop Proportional Gain, or Pos P Gain, to produce a component to the Velocity Command that ultimately attempts to correct for the position error. Too little Pos P Gain results in excessively compliant, or mushy, axis behavior. Too large a Pos P Gain, on the other hand, can result in axis oscillation due to classical servo instability.

To set the gain manually, you must first set the appropriate output scaling factor (either the Velocity Scaling factor or Torque Scaling factor) in the Output tab of this dialog. Your selection of External Drive Configuration type – either Torque or Velocity – in the Servo tab of this dialog determines which scaling factor you must configure before manually setting gains.

If you know the desired loop gain in inches per minute per mil or millimeters per minute per mil, use the following formula to calculate the corresponding P gain:

Pos P Gain = 16.667 * Desired Loop Gain (IPM/mil)

If you know the desired unity gain bandwidth of the position servo in Hertz, use the following formula to calculate the corresponding P gain:

Pos P Gain = Bandwidth (Hertz) * 6.28

The typical value for the Position Proportional Gain is ~100 Sec-1.

Integral (Position) Gain

The Integral (that is, summation) of Position Error is multiplied by the Position Loop Integral Gain, or Pos I Gain, to produce a component to the Velocity Command that ultimately attempts to correct for the position error. Pos I Gain improves the steady-state positioning performance of the system. Increasing the integral gain generally increases the ultimate positioning accuracy of the system. Excessive integral gain, however, results in system instability.

In certain cases, Pos I Gain control is disabled. One such case is when the servo output to the axis' drive is saturated. Continuing integral control behavior in this case would only exacerbate the situation. When the Integrator Hold parameter is set to Enabled, the servo loop automatically disables the integrator during commanded motion.

While the Pos I Gain, if employed, is typically established by the automatic servo tuning procedure (in the Tuning tab of this dialog), the Pos I Gain value may also be set manually. Before doing this it must be stressed that the Output Scaling factor for the axis must be established for the drive system. Once this is done, the Pos I Gain can be computed based on the current or computed value for the Pos P Gain using the following formula:

Pos I Gain = .025 * 0.001 Sec/mSec * (Pos P Gain)2

Assuming a Pos P Gain value of 100 Sec-1 this results in a Pos I Gain value of 2.5 ~0.1 mSec-1 - Sec-1.

Differential

Position Differential Gain helps predict a large overshoot before it happens and makes the appropriate attempt to correct it before the overshoot actually occurs.

Proportional (Velocity) Gain

This parameter is enabled for all loop types except Torque loop.

Velocity Error is multiplied by the Velocity Proportional Gain to produce a component to the Servo Output or Torque Command that ultimately attempts to correct for the velocity error, creating a damping effect. Thus, increasing the Velocity Proportional Gain results in smoother motion, enhanced acceleration, reduced overshoot, and greater system stability. However, too much Velocity Proportional Gain leads to high frequency instability and resonance effects.

If you know the desired unity gain bandwidth of the velocity servo in Hertz, you can use the following formula to calculate the corresponding P gain.

Velocity P Gain = Bandwidth (Hertz) / 6.28

The typical value for the Velocity Proportional Gain is 250.

Integral (Velocity) Gain

This parameter is enabled for all loop types except Torque loop.

At every servo update the current Velocity Error is accumulated in a variable called the Velocity Integral Error. This value is multiplied by the Velocity Integral Gain to produce a component to the Servo Output or Torque Command that attempts to correct for the velocity error. The higher the Vel I Gain value, the faster the axis is driven to the zero Velocity Error condition. Unfortunately, I Gain control is intrinsically unstable. Too much I Gain results in axis oscillation and servo instability.

In certain cases, Vel I Gain control is disabled. One such case is when the servo output to the axis' drive is saturated. Continuing integral control behavior in this case would only exacerbate the situation. When the Integrator Hold parameter is set to Enabled, the servo loop automatically disables the integrator during commanded motion.

Due to the destabilizing nature of Integral Gain, it is recommended that Position Integral Gain and Velocity Integral Gain be considered mutually exclusive. If Integral Gain is needed for the application, use one or the other,

but not both. In general, where static positioning accuracy is required, Position Integral Gain is the better choice.

The typical value for the Velocity Proportional Gain is ~15 mSec-2.

Velocity Feedforward

Velocity Feedforward Gain scales the current Command Velocity by the Velocity Feedforward Gain and adds it as an offset to the Velocity Command. Hence, the Velocity Feedforward Gain allows the following error of the servo system to be reduced to nearly zero when running at a constant speed. This is important in applications such as electronic gearing, position camming, and synchronization applications, where it is necessary that the actual axis position not significantly lag behind the commanded position at any time. The optimal value for Velocity Feedforward Gain is 100%, theoretically. In reality, however, the value may need to be tweaked to accommodate velocity loops with non-infinite loop gain and other application considerations.

Acceleration Feedforward

Acceleration Feedforward Gain scales the current Command Acceleration by the Acceleration Feedforward Gain and adds it as an offset to the Servo Output generated by the servo loop. With this done, the servo loops do not need to generate much of a contribution to the Servo Output, hence the Position and/or Velocity Error values are significantly reduced. Hence, when used in conjunction with the Velocity Feedforward Gain, the Acceleration Feedforward Gain allows the following error of the servo system during the acceleration and deceleration phases of motion to be reduced to nearly zero. This is important in applications such as electronic gearing, position camming, and synchronization applications, where it is necessary that the actual axis position not significantly lag behind the commanded position at any time. The optimal value for Acceleration Feedforward is 100%, theoretically. In reality, however, the value may need to be tweaked to accommodate velocity loops with non-infinite loop gain and other application considerations.

Acceleration Feedforward Gain is not applicable for applications employing velocity loop servo drives. Such systems would require the acceleration feedforward functionality to be located in the drive itself.

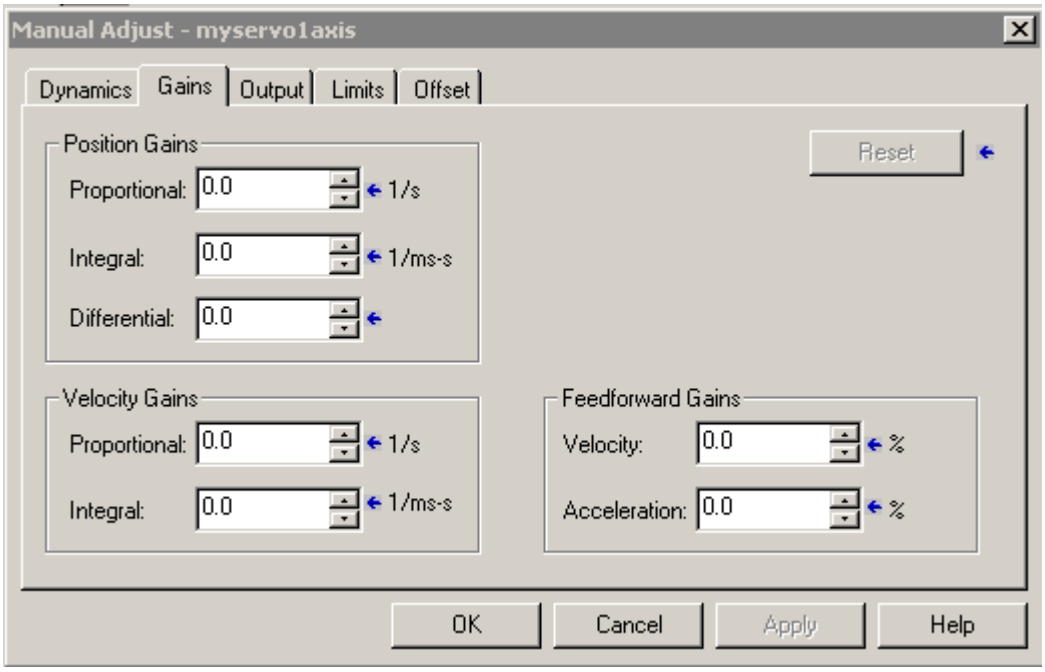
Integrator Hold

If the Integrator Hold parameter is set to:

- Enabled, the servo loop temporarily disables any enabled position or velocity integrators while the command position is changing. This feature is used by point-to-point moves to minimize the integrator wind-up during motion.

- Disabled, all active position or velocity integrators are always enabled.

Manual Adjust Click on this button to access the Gains tab of the Manual Adjust dialog for online editing.

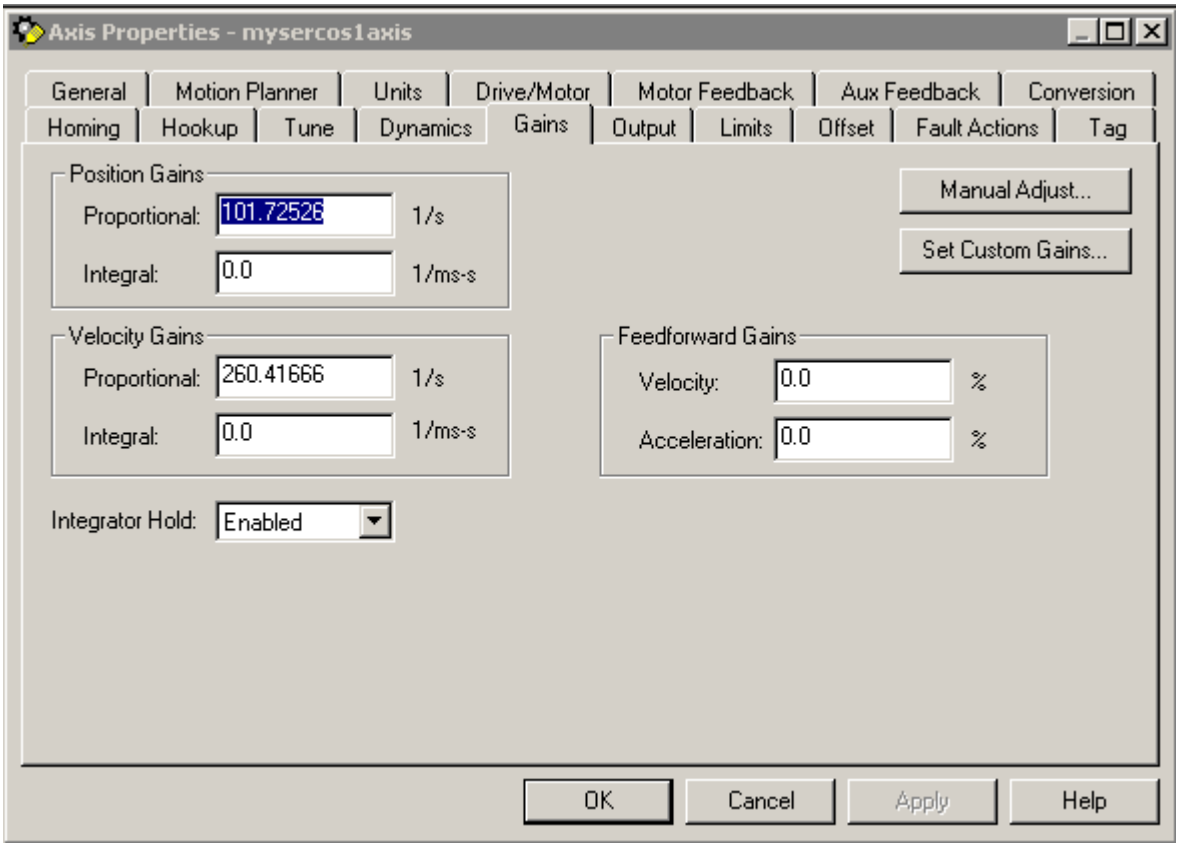


The Manual Adjust button is disabled when RSLogix 5000 software is in Wizard mode, and when you have not yet saved or applied your offline edits to the above parameters.

Gains Tab - AXIS_SERVO_DRIVE

Use this tab to perform the following offline functions:

- Adjust, or tweak gain values that have been automatically set by the tuning process (in the Tune tab of this dialog)
- Manually configure gains for the velocity and position loops
- for an axis of the type AXIS_SERVO_DRIVE.



The drive module uses a nested digital servo control loop consisting of a position loop with proportional, integral and feed-forward gains around an optional digitally synthesized inner velocity loop. The specific design of this nested loop depends upon the Loop Configuration selected in the Drive tab. For a discussion, including a diagram, of a loop configuration, click on the following loop configuration types:

- Motor Position Servo Loop
- Auxiliary Position Servo Loop
- Dual Position Servo Loop
- Motor Dual Command Servo Loop
- Auxiliary Dual Command Servo Loop
- Velocity Servo Loop
- Torque Servo Loop

The parameters on this tab can be edited in either of two ways:

- edit on this tab by typing your parameter changes and then clicking on OK or Apply to save your edits

- edit in the Manual Adjust dialog: click on the Manual Adjust button to open the Manual Adjust dialog to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When RSLogix 5000 software is offline, the following parameters can be edited and the program saved to disk using either the Save command or by clicking on the Apply button. You must re-download the edited program to the controller before it can be run.

Velocity Feedforward

Velocity Feedforward Gain scales the current command velocity (derivative of command position) by the Velocity Feedforward Gain and adds it as an offset to the Velocity Command. Hence, the Velocity Feedforward Gain allows the following error of the servo system to be reduced to nearly zero when running at a constant speed. This is important in applications such as electronic gearing and synchronization applications, where it is necessary that the actual axis position not significantly lag behind the commanded position at any time. The optimal value for Velocity Feedforward Gain is 100%, theoretically. In reality, however, the value may need to be tweaked to accommodate velocity loops with non-infinite loop gain and other application considerations.

Acceleration Feedforward

Acceleration Feedforward Gain scales the current Command Acceleration by the Acceleration Feedforward Gain and adds it as an offset to the Servo Output generated by the servo loop. With this done, the servo loops do not need to generate much of a contribution to the Servo Output, hence the Position and/or Velocity Error values are significantly reduced. Hence, when used in conjunction with the Velocity Feedforward Gain, the Acceleration Feedforward Gain allows the following error of the servo system during the acceleration and deceleration phases of motion to be reduced to nearly zero. This is important in applications such as electronic gearing and synchronization applications, where it is necessary that the actual axis position not significantly lag behind the commanded position at any time. The optimal value for Acceleration Feedforward is 100%, theoretically. In reality, however, the value may need to be tweaked to accommodate velocity loops with non-infinite loop gain and other application considerations.

Acceleration Feedforward Gain is not applicable for applications employing velocity loop servo drives. Such systems would require the acceleration feedforward functionality to be located in the drive itself.

Proportional (Position) Gain

Position Error is multiplied by the Position Loop Proportional Gain, or Pos P Gain, to produce a component to the Velocity Command that ultimately attempts to correct for the position error. Too little Pos P Gain results in excessively compliant, or mushy, axis behavior. Too large a Pos P Gain, on the other hand, can result in axis oscillation due to classical servo instability.

To set the gain manually, you must first set the Torque scaling in the Output tab of this dialog.

If you know the desired loop gain in inches per minute per mil or millimeters per minute per mil, use the following formula to calculate the corresponding P gain:

Pos P Gain = 16.667 * Desired Loop Gain (IPM/mil)

If you know the desired unity gain bandwidth of the position servo in Hertz, use the following formula to calculate the corresponding P gain:

Pos P Gain = Bandwidth (Hertz) * 6.28

The typical value for the Position Proportional Gain is ~100 Sec-1.

Integral (Position) Gain

The Integral (that is, summation) of Position Error is multiplied by the Position Loop Integral Gain, or Pos I Gain, to produce a component to the Velocity Command that ultimately attempts to correct for the position error. Pos I Gain improves the steady-state positioning performance of the system. Increasing the integral gain generally increases the ultimate positioning accuracy of the system. Excessive integral gain, however, results in system instability.

In certain cases, Pos I Gain control is disabled. One such case is when the servo output to the axis' drive is saturated. Continuing integral control behavior in this case would only exacerbate the situation. When the Integrator Hold parameter is set to Enabled, the servo loop automatically disables the integrator during commanded motion.

While the Pos I Gain, if employed, is typically established by the automatic servo tuning procedure (in the Tuning tab of this dialog), the Pos I Gain value may also be set manually. Before doing this it must be stressed that the Torque Scaling factor for the axis must be established for the drive system (in the Output tab of this dialog box). Once this is done, the Pos I Gain can be computed based on the current or computed value for the Pos P Gain using the following formula:

Pos I Gain = .025 * 0.001 Sec/mSec * (Pos P Gain)2

Assuming a Pos P Gain value of 100 Sec-1 this results in a Pos I Gain value of 2.5 ~0.1 mSec-1 - Sec-1.

Proportional (Velocity) Gain

This parameter is enabled only for external drives configured for Torque loop operation in the Servo tab.

Velocity Error is multiplied by the Velocity Proportional Gain to produce a component to the Torque Command that ultimately attempts to correct for the velocity error, creating a damping effect. Thus, increasing the Velocity Proportional Gain results in smoother motion, enhanced acceleration, reduced overshoot, and greater system stability. However, too much Velocity Proportional Gain leads to high frequency instability and resonance effects.

If you know the desired unity gain bandwidth of the velocity servo in Hertz, you can use the following formula to calculate the corresponding P gain.

Vel P Gain = Bandwidth (Hertz) / 6.28

The typical value for the Velocity Proportional Gain is ~250 mSec-1.

Integral (Velocity) Gain

This parameter is enabled only for external drives configured for Torque loop operation in the Servo tab.

At every servo update the current Velocity Error is accumulated in a variable called the Velocity Integral Error. This value is multiplied by the Velocity Integral Gain to produce a component to the Torque Command that attempts to correct for the velocity error. The higher the Vel I Gain value, the faster the axis is driven to the zero Velocity Error condition. Unfortunately, I Gain control is intrinsically unstable. Too much I Gain results in axis oscillation and servo instability.

In certain cases, Vel I Gain control is disabled. One such case is when the servo output to the axis' drive is saturated. Continuing integral control behavior in this case would only exacerbate the situation. When the Integrator Hold parameter is set to Enabled, the servo loop automatically disables the integrator during commanded motion.

Due to the destabilizing nature of Integral Gain, it is recommended that Position Integral Gain and Velocity Integral Gain be considered mutually exclusive. If Integral Gain is needed for the application, use one or the other, but not both. In general, where static positioning accuracy is required, Position Integral Gain is the better choice.

While the Vel I Gain, if employed, is typically established by the automatic servo tuning procedure (in the Tune tab of this dialog box), the Pos I Gain value may also be set manually. Before doing this it must be stressed that the Torque Scaling factor for the axis must be established for the drive system, in the Output tab. Once this is done the Vel I Gain can be computed based on the current or computed value for the Vel P Gain using the following formula:

Vel I Gain = 0.25 * 0.001 Sec/mSec * (Vel P Gain)2

The typical value for the Velocity Proportional Gain is ~15 mSec-2.

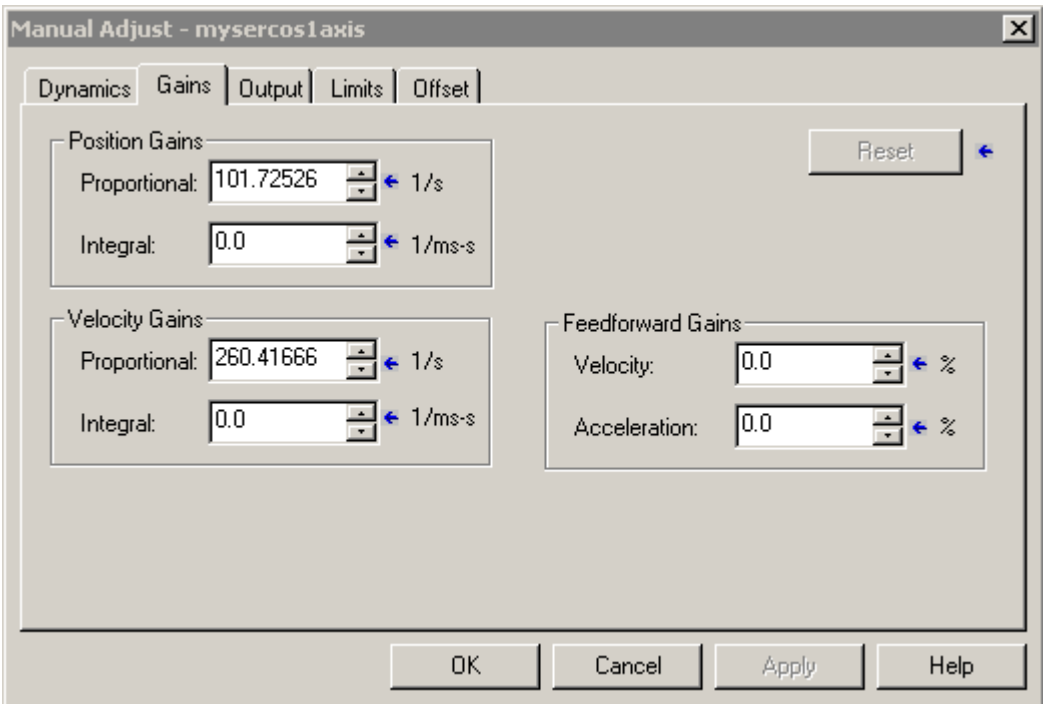
Integrator Hold

If the Integrator Hold parameter is set to:

- Enabled, the servo loop temporarily disables any enabled position or velocity integrators while the command position is changing. This feature is used by point-to-point moves to minimize the integrator wind-up during motion.
- Disabled, all active position or velocity integrators are always enabled.

Manual Adjust

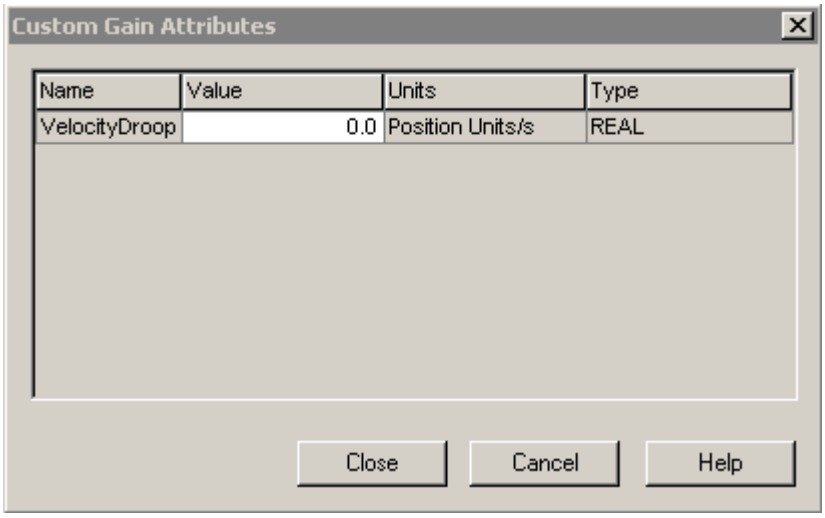
Click on this button to access the Gains tab of the Manual Adjust dialog for online editing.



The Manual Adjust button is disabled when RSLogix 5000 software is in Wizard mode, and when you have not yet saved or applied your offline edits to the above parameters.

Set Custom Gains

Click on this button to open the Custom Gain Attributes dialog.



At this dialog box you can edit the VelocityDroop attribute.

When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

When multiple workstations connect to the same controller using RSLogix 5000 software and invoke the Axis Wizard or Axis Properties dialog, the firmware allows only the first workstation to make any changes to axis attributes. The second workstation switches to a Read Only mode, indicated in the title bar, so that you may view the changes from that workstation, but not edit them.

Attribute

The following attribute value can be monitored and edited in this dialog box.

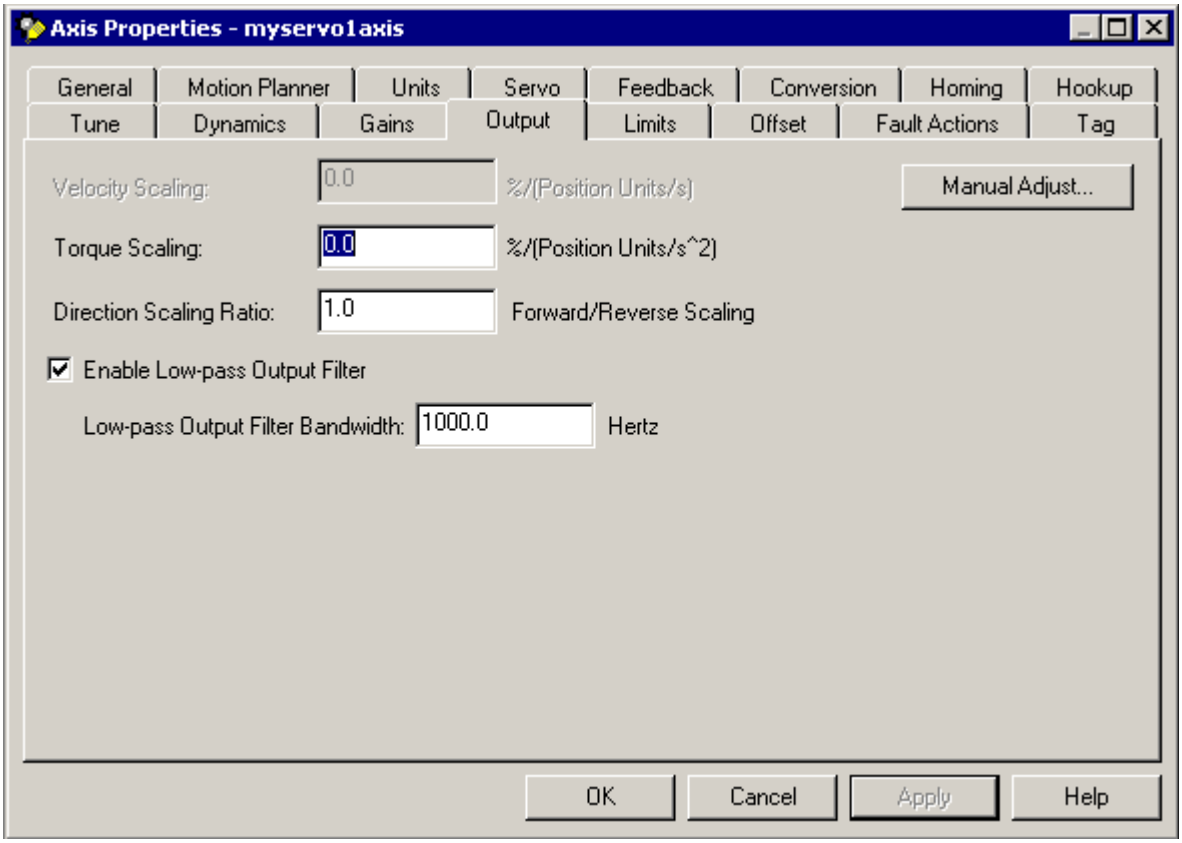
Attribute	Description
VelocityDroop	This 32-bit unsigned attribute – also referred to as static gain – acts as a very slow discharge of the velocity loop integrator. VelocityDroop may be used as a component of an external position loop system where setting this parameter to a higher, nonzero value eliminates servo hunting due to load/stick friction effects. This parameter only has effect if VelocityIntegralGain is not zero. Its value ranges from 0 to 2.14748x10^12. This value is not applicable for Ultra3000 drives.

Output Tab - AXIS_SERVO

Use this dialog for offline configuration of:

- scaling values, which are used to generate gains, and
- the servo's low-pass digital output filter

for an axis of the type AXIS_SERVO configured as a Servo drive in the General tab of this dialog.



The parameters on this tab can be edited in either of two ways:

- edit on this tab by typing your parameter changes and then clicking on OK or Apply to save your edits

- edit in the Manual Adjust dialog: click on the Manual Adjust button to open the Manual Adjust dialog to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When RSLogix 5000 software is offline, the following parameters can be edited and the program saved to disk using either the Save command or by clicking on the Apply button. You must re-download the edited program to the controller before it can be run.

Velocity Scaling

The Velocity Scaling attribute is used to convert the output of the servo loop into equivalent voltage to an external velocity servo drive. This has the effect of "normalizing" the units of the servo loop gain parameters so that their values are not affected by variations in feedback resolution, drive scaling, or mechanical gear ratios. The Velocity Scaling value is typically established by servo's automatic tuning procedure but these values can be calculated, if necessary, using the following guidelines.

If the axis is configured for a velocity external servo drive (in the Servo tab of this dialog), the software velocity loop in the servo module is disabled. In this case the Velocity Scaling value can be calculated by the following formula:

Velocity Scaling = 100% / (Speed @ 100%)

For example, if this axis is using position units of motor revolutions (revs), and the servo drive is scaled such that with an input of 100% (for example, 10 Volts) the motor goes 5,000 RPM (or 83.3 RPS), the Velocity Scaling attribute value would be calculated as:

Velocity Scaling = 100% / (83.3 RPS) = 1.2% / Revs Per Second

Torque Scaling

The Torque Scaling attribute is used to convert the acceleration of the servo loop into equivalent % rated torque to the motor. This has the effect of "normalizing" the units of the servo loops gain parameters so that their values are not affected by variations in feedback resolution, drive scaling, motor and load inertia, and mechanical gear ratios. The Torque Scaling value is typically established by the controller's automatic tuning procedure but the value can be manually calculated, if necessary, using the following guidelines:

Torque Scaling = 100% Rated Torque / (Acceleration @ 100% Rated Torque)

For example, if this axis is using position units of motor revolutions (revs), with 100% rated torque applied to the motor, if the motor accelerates at a rate of 3000 Revs/Sec2, the Torque Scaling attribute value would be calculated as shown below:

Torque Scaling = 100% Rated / (3000 RPS2) = 0.0333% Rated/ Revs Per Second2

If the Torque Scaling value does not reflect the true torque to acceleration characteristic of the system, the gains also does not reflect the true performance of the system.

Enable Low-pass Output Filter

Select this to enable the servo's low-pass digital output filter. De-select this to disable this filter.

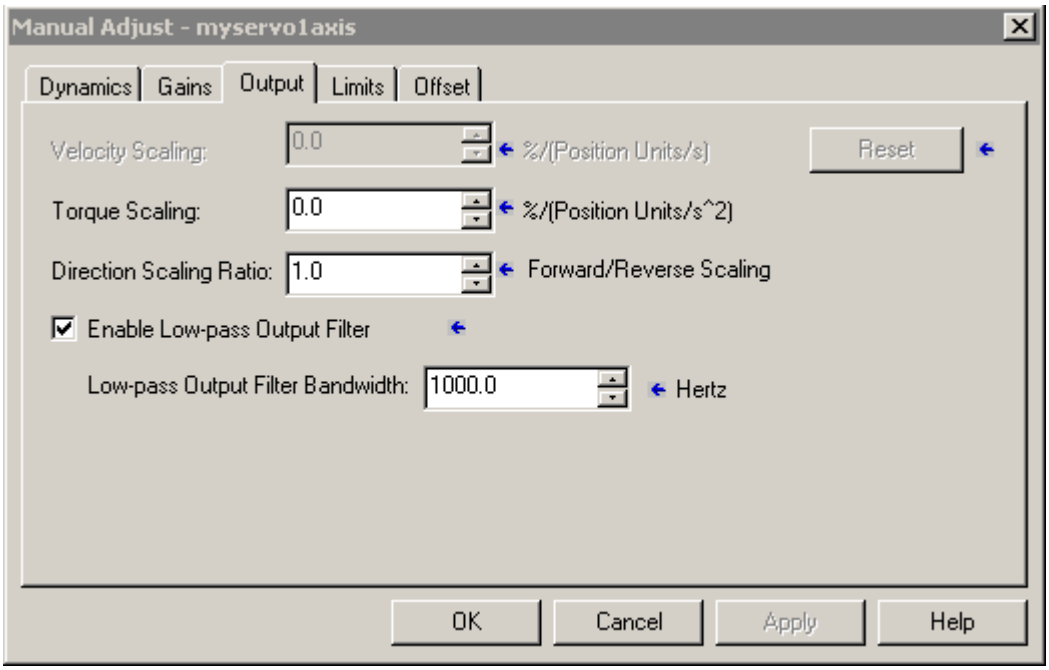
During tuning, if the controller detects a high degree of tuning inertia, it enables the Low Pass Output Filter and calculates and sets a value for Low Pass Output Filter Bandwidth.

Low-pass Output Filter Bandwidth

With Enable Low-pass Output Filter selected, this value sets the bandwidth, in Hertz, of the servo's low-pass digital output filter. Use this output filter to filter out high frequency variation of the servo module output to the drive. All output from the servo module greater than the Filter Bandwidth setting is filtered-out, and not sent to the drive.

If the Low-pass Output Filter Bandwidth value is set to zero, the low-pass output filter is disabled. The lower the Filter Bandwidth value, the greater the attenuation of these high frequency components of the output signal. Because the low-pass filter adds lag to the servo loop, which pushes the system towards instability, decreasing the Filter Bandwidth value usually requires lowering the Position or Velocity Proportional Gain settings to maintain stability. The output filter is particularly useful in high inertia applications where resonance behavior can severely restrict the maximum bandwidth capability of the servo loop.

Manual Adjust Click on this button to access the Output tab of the Manual Adjust dialog for online editing.



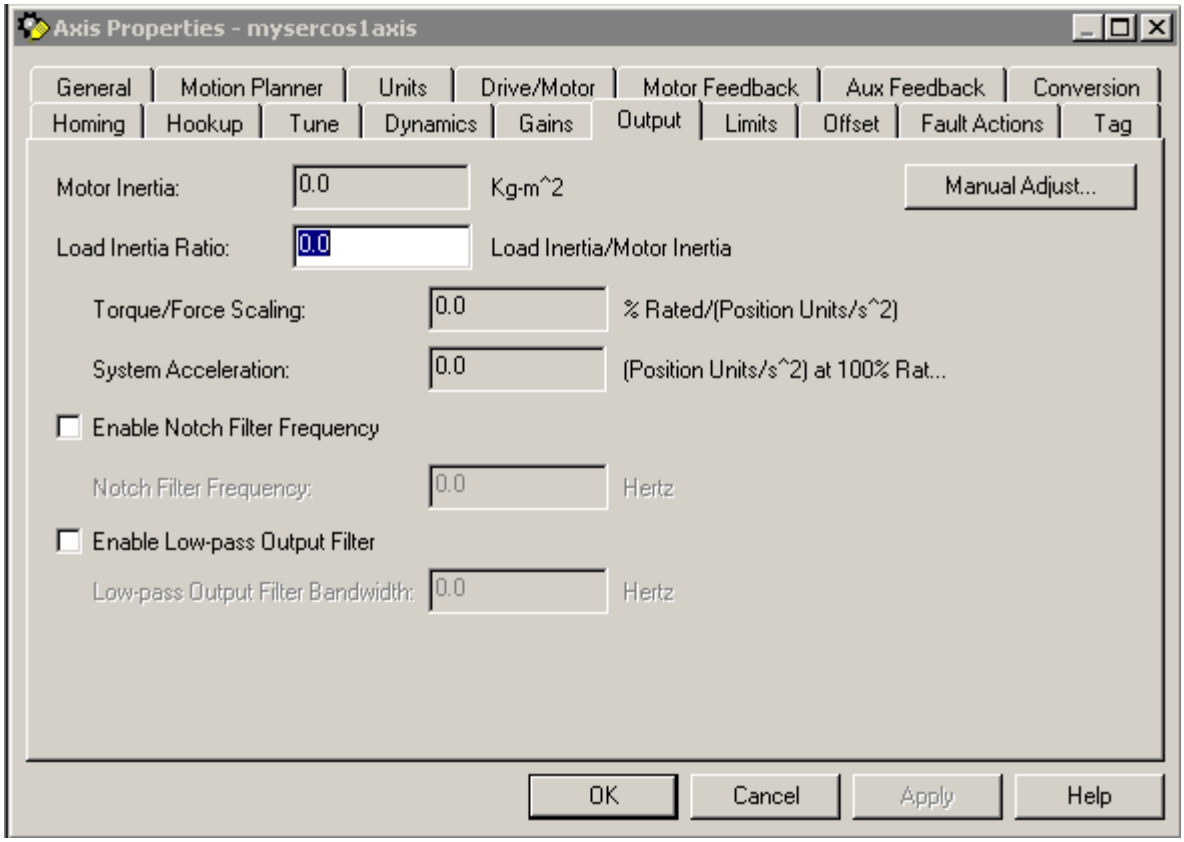
The Manual Adjust button is disabled when RSLogix 5000 software is in Wizard mode, and when you have not yet saved or applied your offline edits to the above parameters.

Output Tab Overview - AXIS_SERVO_DRIVE

Use this dialog box to make the following offline configurations:

- set the torque scaling value, which is used to generate gains
- enable and configure the Notch Filter
- enable and configure servo's low-pass digital output filter

for an axis of the type AXIS_SERVO_DRIVE, configured as a Servo drive in the General tab of this dialog.



The parameters on this tab can be edited in either of two ways:

- edit on this tab by typing your parameter changes and then clicking on OK or Apply to save your edits
- edit in the Manual Adjust dialog: click on the Manual Adjust button to open the Manual Adjust dialog to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When RSLogix 5000 software is offline, the following parameters can be edited and the program saved to disk using either the Save command or by clicking on the Apply button. You must re-download the edited program to the controller before it can be run.

Motor Inertia

The Motor Inertia value represents the inertia of the motor without any load attached to the motor shaft in Torque Scaling units.

Load Inertia Ratio

The Load Inertia Ratio value represents the ratio of the load inertia to the motor inertia.

Torque Scaling

The Torque Scaling attribute is used to convert the acceleration of the servo loop into equivalent % rated torque to the motor. This has the effect of normalizing the units of the servo loops gain parameters so that their values are not affected by variations in feedback resolution, drive scaling, motor and load inertia, and mechanical gear ratios. The Torque Scaling value is typically established by the controller's automatic tuning procedure but the value can be manually calculated, if necessary, using the following guidelines:

Torque Scaling = 100% Rated Torque / (Acceleration @ 100% Rated Torque)

For example, if this axis is using position units of motor revolutions (revs), with 100% rated torque applied to the motor, if the motor accelerates at a rate of 3000 Revs/Sec2, the Torque Scaling attribute value would be calculated as shown below:

Torque Scaling = 100% Rated / (3000 RPS2) = 0.0333% Rated/ Revs Per Second2

If the Torque Scaling value does not reflect the true torque to acceleration characteristic of the system, the gains also do not reflect the true performance of the system.

Enable Notch Filter

Select this to enable the drive's notch filter. De-select this to disable this filter.

Notch Filter

With Enable Notch Filter selected, this value sets the center frequency of the drive's digital notch filter. If the Notch Filter value is set to zero, the notch filter is disabled.

Currently implemented as a 2nd order digital filter with a fixed Q, the Notch Filter provides approximately 40DB of output attenuation at the Notch Filter frequency. This output notch filter is particularly useful in attenuating mechanical resonance phenomena. The output filter is particularly useful in high inertia applications where mechanical resonance behavior can severely restrict the maximum bandwidth capability of the servo loop.

This value is not applicable for Ultra3000 drives.

Enable Low-pass Output Filter

Select this to enable the servo's low-pass digital output filter. De-select this to disable this filter.

During tuning, if the controller detects a high degree of tuning inertia, the controller enables the Low Pass Output Filter and calculates and sets a value for Low Pass Output Filter Bandwidth.

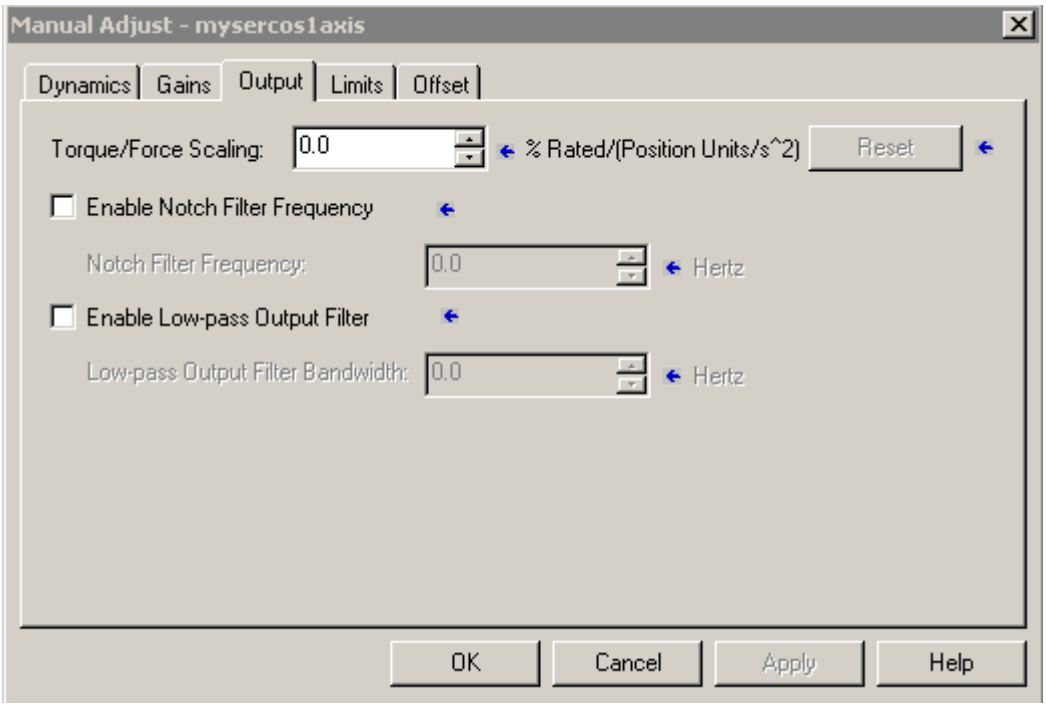
Low-pass Output Filter Bandwidth

With Enable Low-pass Output Filter selected, this value sets the bandwidth, in Hertz, of the servo's low-pass digital output filter. Use this output filter to filter out high frequency variation of the servo module output to the drive. All output from the servo module greater than the Filter Bandwidth setting is filtered-out, and not sent to the drive.

If the Low-pass Output Filter Bandwidth value is set to zero, the low-pass output filter is disabled. The lower the Filter Bandwidth value, the greater the attenuation of these high frequency components of the output signal. Because the low-pass filter adds lag to the servo loop, which pushes the system towards instability, decreasing the Filter Bandwidth value usually requires lowering the Position or Velocity Proportional Gain settings to maintain stability. The output filter is particularly useful in high inertia applications where resonance behavior can severely restrict the maximum bandwidth capability of the servo loop.

Manual Adjust

Click on this button to open the Output tab of the Manual Adjust dialog for online editing of Torque/Force Scaling, the Notch Filter Frequency, and the Low-pass Output Filter parameters.



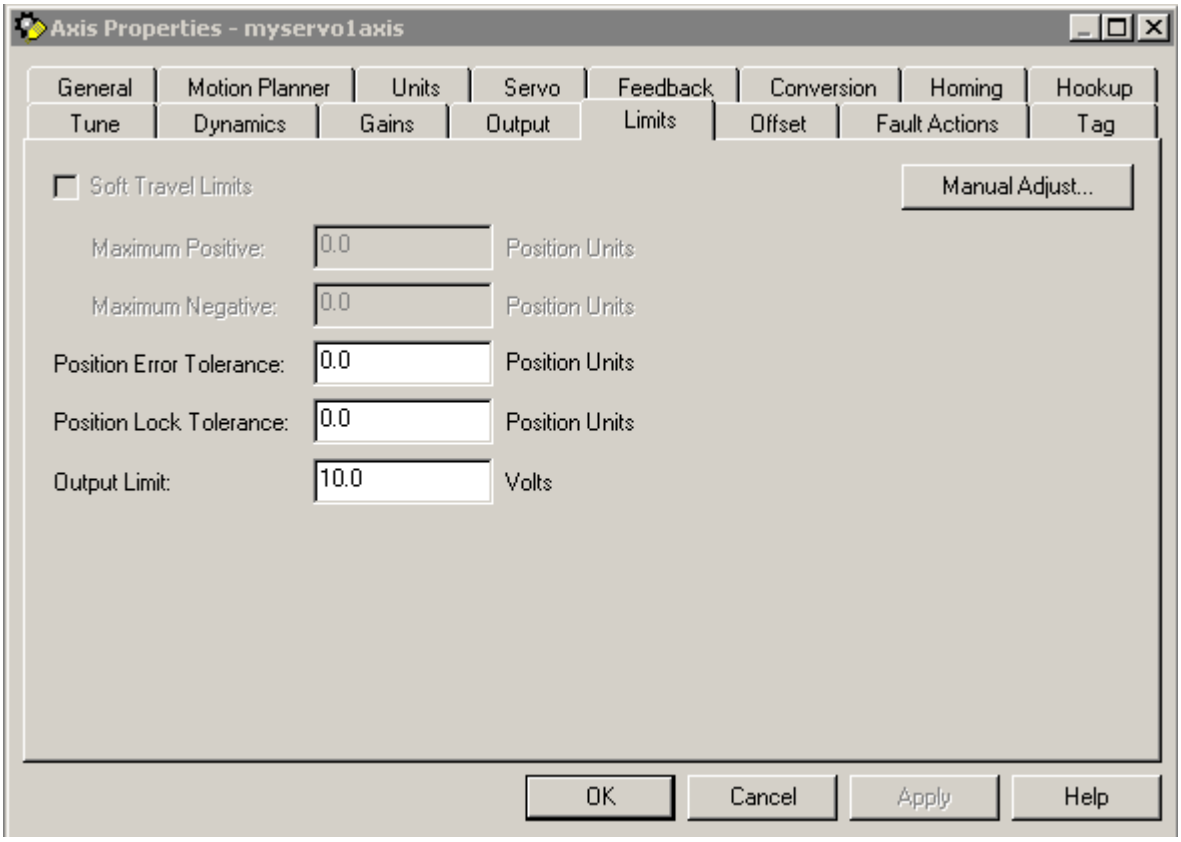
The Manual Adjust button is disabled when RSLogix 5000 software is in Wizard mode, and when offline edits to the above parameters have not yet been saved or applied.

Limits Tab - AXIS_SERVO

Use this tab to make the following offline configurations:

- enable and set maximum positive and negative software travel limits, and
- configure both Position Error Tolerance and Position Lock Tolerance, and
- set the servo drive's Output Limit

for an axis of the type AXIS_SERVO configured as a Servo drive in the General tab of this dialog.



The parameters on this tab can be edited in either of two ways:

- edit on this tab by typing your parameter changes and then clicking on OK or Apply to save your edits
- edit in the Manual Adjust dialog: click on the Manual Adjust button to open the Manual Adjust dialog to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When RSLogix 5000 software is offline, the following parameters can be edited and the program saved to disk using either the Save command or by clicking on the Apply button. You must re-download the edited program to the controller before it can be run.

Soft Travel Limits

Enables software overtravel checking for an axis when Positioning Mode is set to Linear (in the Conversion tab of this dialog). If an axis is configured for software overtravel limits and if that axis passes beyond these maximum travel limits (positive or negative), a software overtravel fault is issued. The response to this fault is specified by the Soft Overtravel setting (in the Fault Actions tab of this dialog). Software overtravel limits are disabled during the tuning process.

Maximum Positive

Type the maximum positive position to be used for software overtravel checking, in position units.

The Maximum Positive limit must always be greater than the Maximum Negative limit.

Maximum Negative

Type the maximum negative position to be used for software overtravel checking, in position units.

The Maximum Negative limit must always be less than the Maximum Positive limit.

Position Error Tolerance

Specifies how much position error the servo tolerates before issuing a position error fault. This value is interpreted as a +/- quantity.

For example, setting Position Error Tolerance to 0.75 position units means that a position error fault is generated whenever the position error of the axis is greater than 0.75 or less than -0.75 position units, as shown here:

This value is set to twice the following error at maximum speed based on the measured response of the axis, during the autotuning process. In most applications, this value provides reasonable protection in case of an axis fault or stall condition without nuisance faults during normal operation. If you need to change the calculated position error tolerance value, the recommended setting is 150% to 200% of the position error while the axis is running at its maximum speed.

Position Lock Tolerance

Specifies the maximum position error the servo module accepts in order to indicate the Position Lock status bit is set. This is useful in determining when the desired end position is reached for position moves. This value is interpreted as a +/- quantity.

For example, specifying a lock tolerance of 0.01 provides a minimum positioning accuracy of +/- 0.01 position units, as shown here:

Output Limit

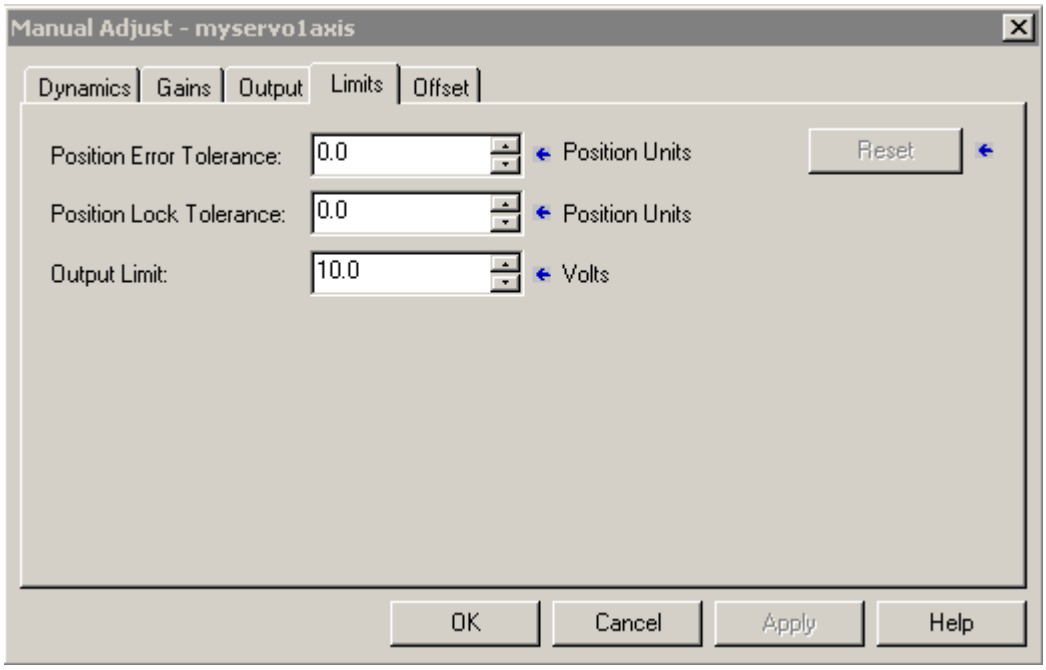
Provides a method of limiting the maximum servo output voltage of a physical axis to a specified level. The servo output for the axis as a function of position servo error, both with and without servo output limiting, is shown below.

The servo output limit may be used as a software current or torque limit if you are using a servo drive in torque loop mode. The percentage of the drive's maximum current that the servo controller ever commands is equal to the specified servo output limit. For example, if the drive is capable of 30 Amps of current for a 10 Volt input, setting the servo output limit to 5V limits the maximum drive current to 15 Amps.

The servo output limit may also be used if the drive cannot accept the full ±10 Volt range of the servo output. In this case, the servo output limit value effectively limits the maximum command sent to the amplifier. For example, if the drive can only accept command signals up to ±7.5 Volts, set the servo output limit value to 7.5 volts.

Manual Adjust

Click on this button to open the Limits tab of the Manual Adjust dialog for online editing of the Position Error Tolerance, Position Lock Tolerance, and Output Limit parameters.



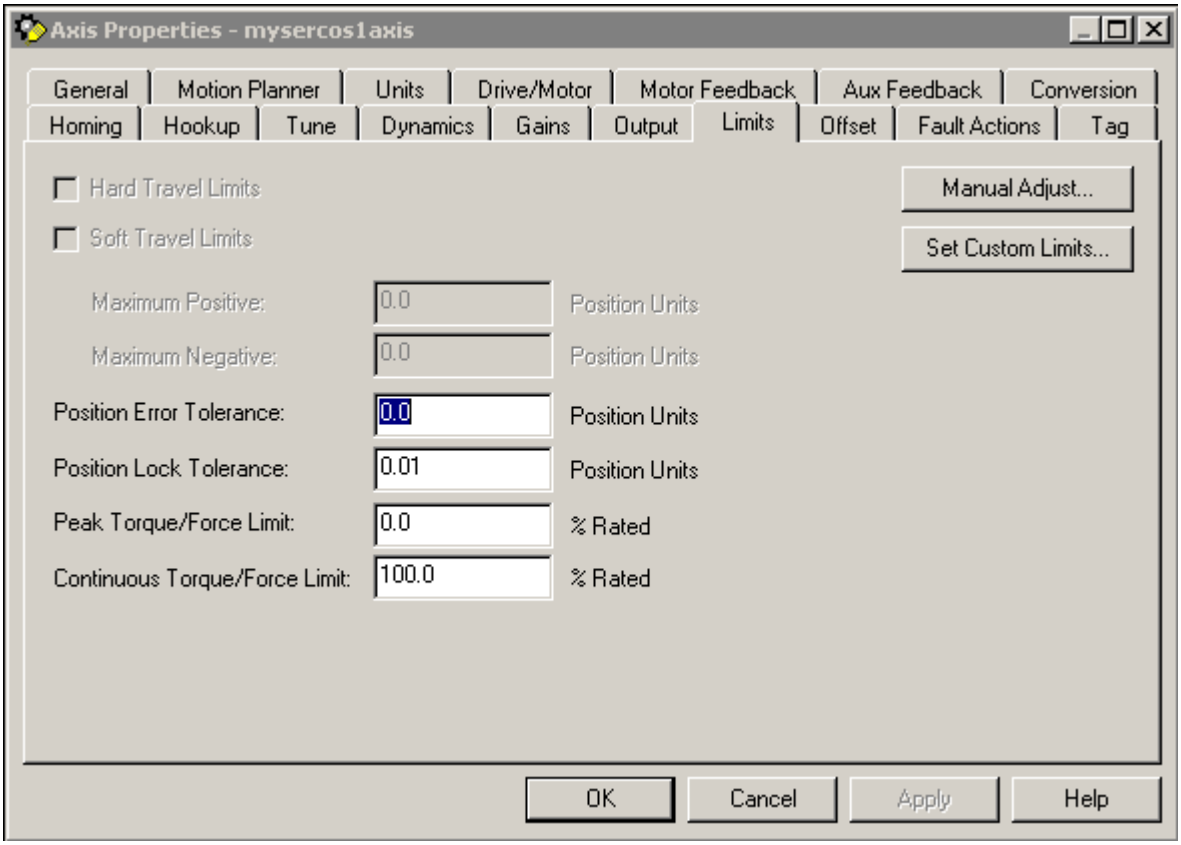
The Manual Adjust button is disabled when RSLogix 5000 software is in Wizard mode, and when offline edits to the above parameters have not yet been saved or applied.

Limits Tab - AXIS_SERVO_DRIVE

Use this tab to make the following offline configurations:

- enable and set maximum positive and negative software travel limits, and
- configure both Position Error Tolerance and Position Lock Tolerance,

for an axis of the type AXIS_SERVO_DRIVE configured as a Servo drive in the General tab of this dialog.



The parameters on this tab can be edited in either of two ways:

- edit on this tab by typing your parameter changes and then clicking on OK or Apply to save your edits

- edit in the Manual Adjust dialog: click on the Manual Adjust button to open the Manual Adjust dialog to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When RSLogix 5000 software is offline, the following parameters can be edited and the program saved to disk using either the Save command or by

clicking on the Apply button. You must re-download the edited program to the controller before it can be run.

Hard Travel Limits

Enables a periodic test that monitors the current state of the positive and negative overtravel limit switch inputs, when Positioning Mode is set to Linear (in the Conversion tab of this dialog). If an axis is configured for hardware overtravel checking and if that axis passes beyond a positive or negative overtravel limit switch, a Positive Hard Overtravel Fault or Negative Hard Overtravel Fault is issued. The response to this fault is specified by the Hard Overtravel setting (in the Fault Actions tab of this dialog).

Soft Travel Limits

Enables software overtravel checking for an axis when Positioning Mode is set to Linear (in the Conversion tab of this dialog). If an axis is configured for software overtravel limits and if that axis passes beyond these maximum travel limits (positive or negative), a software overtravel fault is issued. The response to this fault is specified by the Soft Overtravel setting (in the Fault Actions tab of this dialog). Software overtravel limits are disabled during the tuning process.

Maximum Positive

Type the maximum positive position to be used for software overtravel checking, in position units.

The Maximum Positive limit must always be greater than the Maximum Negative limit.

Maximum Negative

Type the maximum negative position to be used for software overtravel checking, in position units.

The Maximum Negative limit must always be less than the Maximum Positive limit.

Position Error Tolerance

Specifies how much position error the servo tolerates before issuing a position error fault. This value is interpreted as a +/- quantity.

For example, setting Position Error Tolerance to 0.75 position units means that a position error fault is generated whenever the position error of the axis is greater than 0.75 or less than -0.75 position units, as shown here:

This value is set to twice the following error at maximum speed based on the measured response of the axis, during the autotuning process. In most applications, this value provides reasonable protection in case of an axis fault

or stall condition without nuisance faults during normal operation. If you need to change the calculated position error tolerance value, the recommended setting is 150% to 200% of the position error while the axis is running at its maximum speed.

Position Lock Tolerance

Specifies the maximum position error the servo module accepts in order to indicate the Position Lock status bit is set. This is useful in determining when the desired end position is reached for position moves. This value is interpreted as a +/- quantity.

For example, specifying a lock tolerance of 0.01 provides a minimum positioning accuracy of +/- 0.01 position units, as shown here:

Peak Torque/Force Limit

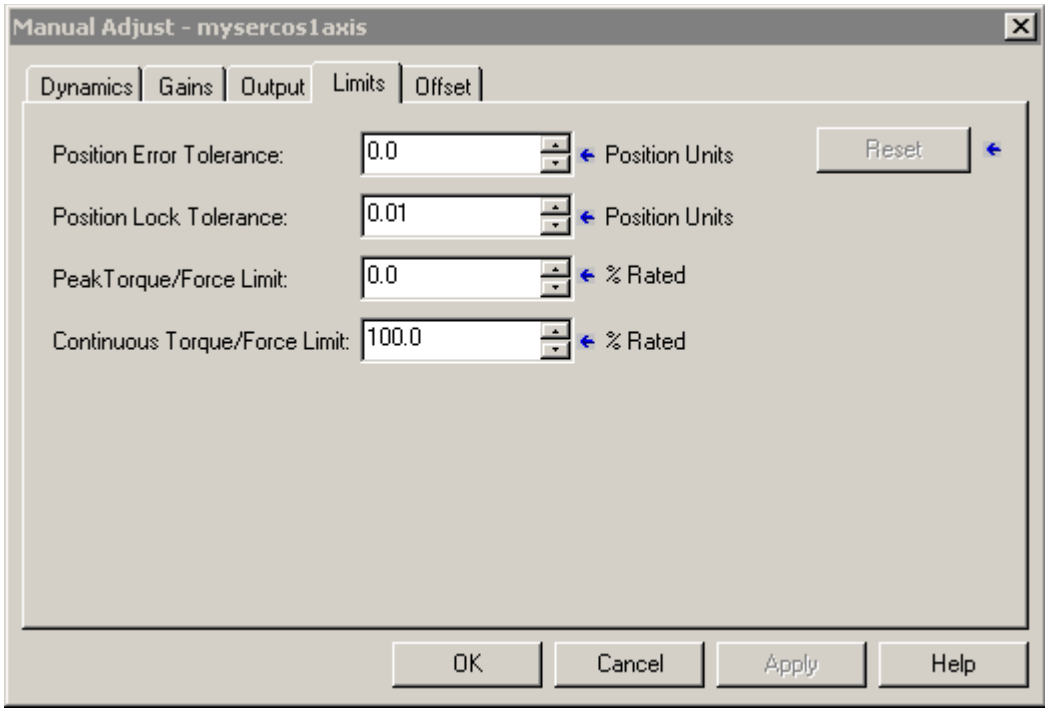
The Peak Torque/Force Limit specifies the maximum percentage of the motors rated current that the drive can command as either positive or negative torque/force. For example, a torque limit of 150% shall limit the current delivered to the motor to 1.5 times the continuous current rating of the motor.

Continuous Torque/Force Limit

The Continuous Torque/Force Limit specifies the maximum percentage of the motors rated current that the drive can command on a continuous or RMS basis. For example, a Continuous Torque/Force Limit of 150% limits the continuous current delivered to the motor to 1.5 times the continuous current rating of the motor.

Manual Adjust

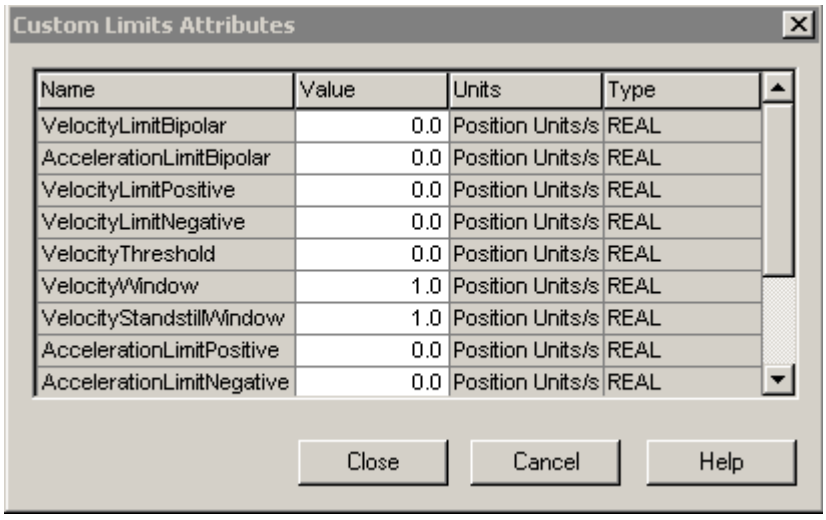
Click on this button to open the Limits tab of the Manual Adjust dialog for online editing of the Position Error Tolerance, Position Lock Tolerance, Peak Torque/Force Limit, and Continuous Torque/Force Limit parameters.



The Manual Adjust button is disabled when RSLogix 5000 software is in Wizard mode, and when offline edits to the above parameters have not yet been saved or applied.

Set Custom Limits

Click this button to open the Custom Limit Attributes dialog.



From this dialog box you can monitor and edit the limit-related attributes.

When RSLogix 5000 software is online, the parameters on this tab transition to a read-only state. When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

When multiple workstations connect to the same controller using RSLogix 5000 software and invoke the Axis Wizard or Axis Properties dialog, the firmware allows only the first workstation to make any changes to axis attributes. The second workstation switches to a Read Only mode, indicated in the title bar, so that you may view the changes from that workstation, but not edit them.

Attributes

The following attribute values can be monitored and edited in this dialog box.

Attribute	Description
VelocityLimitBipolar	This attribute sets the velocity limit symmetrically in both directions. If the command velocity exceeds this value, VelocityLimitStatusBit of the DriveStatus attribute is set. This attribute has a value range of 0 to $2.14748x10^{12}$.
AccelerationLimitBipolar	This attribute sets the acceleration and deceleration limits for the drive. If the command acceleration exceeds this value, AccelLimitStatusBit of the DriveStatus attribute is set. This attribute has a value range of 0 to $2.14748x10^{15}$.
TorqueLimitBipolar	This attribute sets the torque limit symmetrically in both directions. When actual torque exceeds this value TorqueLimitStatus of the DriveStatus attribute is set. This attribute has a value range of 0 to 1000.
VelocityLimitPositive	This attribute displays the maximum allowable velocity in the positive direction. If the velocity limit is exceeded, bit 5 ("Velocity Command Above Velocity Limit") VelocityLimitStatusBit of the DriveStatus attribute is set. This attribute has a value range of 0 to $2.14748x10^{12}$.
VelocityLimitNegative	This attribute displays the maximum allowable velocity in the negative direction. If the velocity limit is exceeded, bit 5 ("Velocity Command Above Velocity Limit") VelocityLimitStatusBit of the DriveStatus attribute is set. This attribute has a value range of $-2.14748x10^{12}$ to 0.

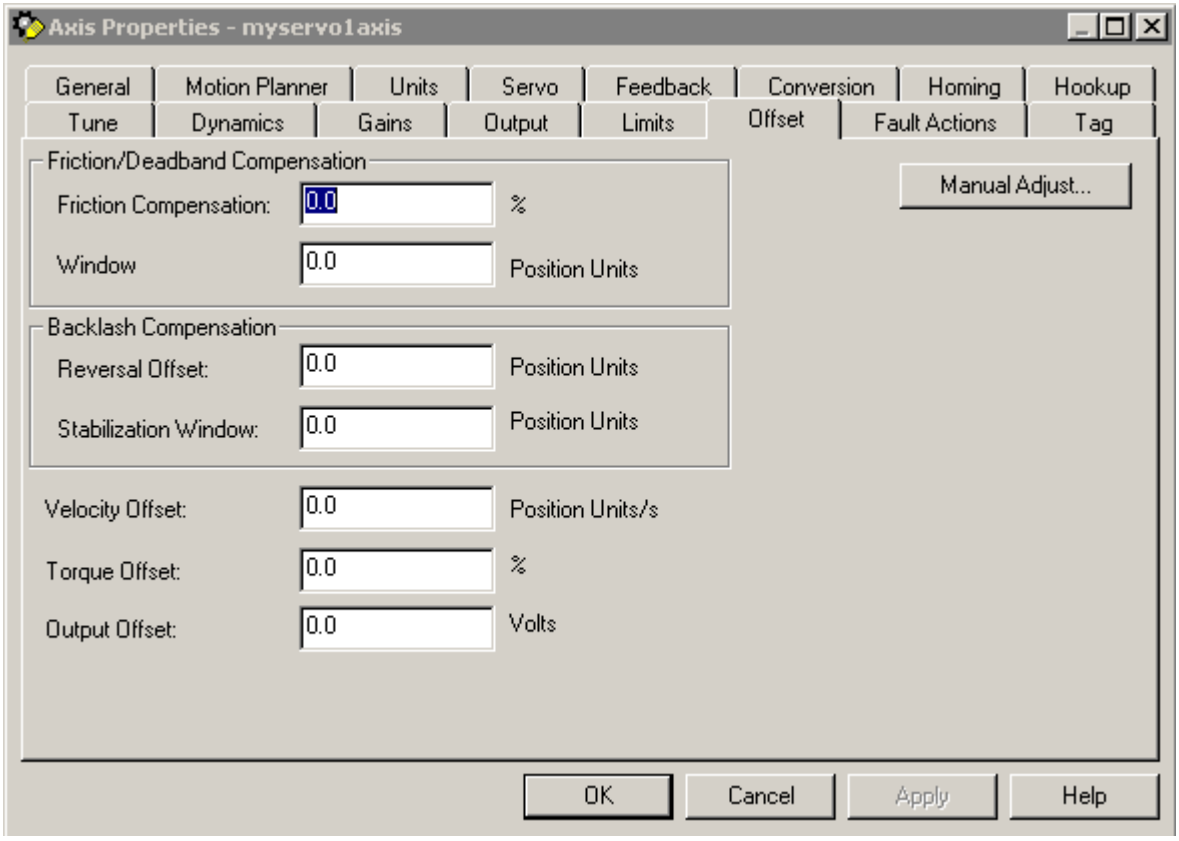
Attribute	Description
VelocityThreshold	This attribute displays the velocity threshold limit. If the motor velocity is less than this limit, VelocityThresholdStatus of the DriveStatus attribute is set. This attribute has a value range of 0 to $2.14748x10^{12}$.
VelocityWindow	This attribute displays the limits of the velocity window. If the motor's actual velocity differs from the command velocity by an amount less that this limit VelocityLockStatus of the DriveStatus attribute is set. This attribute has a value range of 0 to $2.14748x10^{12}$.
VelocityStandstillWindow	This attribute displays the velocity limit for the standstill window. If the motor velocity is less than this limit VelocityStandStillStatus of the DriveStatus bit is set. This attribute has a value range of 0 to $2.14748x10^{12}$.
AccelerationLimitPositive	This attribute limits the maximum acceleration ability of the drive to the programmed value. If the command acceleration exceeds this value, AccelLimitStatusBit of the DriveStatus attribute is set. This attribute has a value range of 0 to $2.14748x10^{15}$.
AccelerationLimitNegative	This attribute limits the maximum acceleration ability of the drive to the programmed value. If the command acceleration exceeds this value, the AccelLimitStatus bit of the DriveStatus attribute is set. This attribute has a value range of $-2.14748x10^{15}$ to 0.
TorqueLimitPositive	This attribute displays the maximum torque in the positive direction. If the torque limit is exceeded, the TorqueLimitStatus bit of the DriveStatus attribute is set. This attribute has a value range of 0 to 1000.
TorqueLimitNegative	This attribute displays the maximum torque in the negative direction. If the torque limit is exceeded, the TorqueLimitStatus bit of the DriveStatus attribute is set. This attribute has a value range of -1000 to 0.
TorqueThreshold	This attribute displays the torque threshold. If this limit is exceeded, the TorqueThreshold bit of the DriveStatus attribute is set. This attribute has a value range of 0 to 1000.

Offset Tab - AXIS_SERVO

Use this tab to make offline adjustments to the following Servo Output values:

- Friction Compensation
- Velocity Offset
- Torque Offset
- Output Offset

for an axis of the type AXIS_SERVO configured as a Servo drive in the General tab of this dialog.



The parameters on this tab can be edited in either of two ways:

- edit on this tab by typing your parameter changes and then clicking on OK or Apply to save your edits

- edit in the Manual Adjust dialog: click on the Manual Adjust button to open the Manual Adjust dialog to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When RSLogix 5000 software is offline, the following parameters can be edited and the program saved to disk using either the Save command or by clicking on the Apply button. You must re-download the edited program to the controller before it can be run.

Friction/Deadband Compensation

Friction Compensation

The percentage of output level added to a positive current Servo Output value, or subtracted from a negative current Servo Output value, for the purpose of moving an axis that is stuck in place due to static friction.

It is not unusual for an axis to have enough static friction (called "sticktion") that, even with a significant position error, the axis refuses to budge. Friction Compensation is used to break "sticktion" in the presence of a nonzero position error. This is done by adding, or subtracting, a percentage output level), called Friction Compensation to the Servo Output value.

The Friction Compensation value should be just less than the value that would break the "sticktion"

A larger value can cause the axis to "dither", that is, move rapidly back and forth about the commanded position.

Friction Compensation Window

To address the issue of dither when applying Friction Compensation and hunting from the integral gain, a Friction Compensation Window is applied around the current command position when the axis is not being commanded to move. If the actual position is within the Friction Compensation Window the Friction Compensation value is applied to the Servo Output but scaled by the ratio of the position error to the Friction Compensation Window. Within the window, the servo integrators are also disabled. Thus, once the position error reaches or exceeds the value of the Friction Compensation Window attribute, the full Friction Compensation value is applied. If the Friction Compensation Window is set to zero, this feature is effectively disabled.

A nonzero Friction Compensation Window has the effect of softening the Friction Compensation as its applied to the Servo Output and reducing the dithering effect that it can create. This generally allows higher values of Friction Compensation to be applied. Hunting is also eliminated at the cost of a small steady-state error.

Backlash Compensation

Reversal Offset

Backlash Reversal Offset provides the capability to compensate for positional inaccuracy introduced by mechanical backlash. For example, power-train type applications require a high level of accuracy and repeatability during machining operations. Axis motion is often generated by a number of mechanical components, a motor, a gearbox, and a ball-screw that may introduce inaccuracies and that are subject to wear over their lifetime. Therefore, when an axis is commanded to reverse direction, mechanical play in the machine (through the gearing, ball-screw, and so on) may result in a small amount of motor motion without axis motion. As a result, the feedback device may indicate movement even though the axis has not physically moved.

If a value of zero is applied to the Backlash Reversal Offset, the feature is effectively disabled. Once enabled by a nonzero value, and the load is engaged by a reversal of the commanded motion, changing the Backlash Reversal Offset can cause the axis to shift as the offset correction is applied to the command position.

Stabilization Window

The Backlash Stabilization Window controls the Backlash Stabilization feature in the servo control loop.

Properly configured with a suitable value for the Backlash Stabilization Window, entirely eliminates the gearbox buzz without sacrificing any servo performance. In general, this value should be set to the measured backlash distance. A Backlash Stabilization Window value of zero effectively disables the feature.

Velocity Offset

Provides a dynamic velocity correction to the output of the position servo loop, in position units per second.

Torque Offset

Provides a dynamic torque command correction to the output of the velocity servo loop, as a percentage of velocity servo loop output.

Output Offset

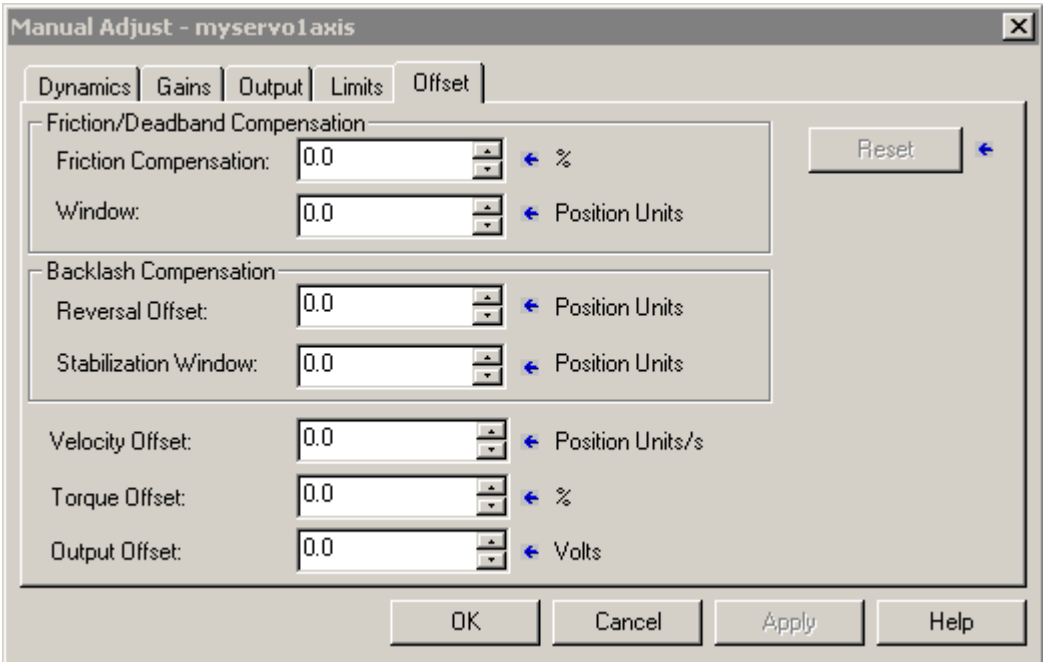
Corrects the problem of axis “drift”, by adding a fixed voltage value (not to exceed ±10 Volts) to the Servo Output value. Input a value to achieve near zero drive velocity when the uncompensated Servo Output value is zero.

When interfacing an external Servo Drive – especially for velocity servo drives, it is necessary to compensate for the effect of drive offset. Cumulative offsets

of the servo module's DAC output and the Servo Drive Input result in a situation where a zero commanded Servo Output value causes the axis to "drift". If the drift is excessive, it can cause problems with the Hookup Diagnostic and Tuning procedures, as well as result in a steady-state nonzero position error when the servo loop is closed.

Manual Adjust

Click on this button to open the Offset tab of the Manual Adjust dialog for online editing of the Friction/Deadband Compensation, Backlash Compensation, Velocity Offset, Torque Offset, and Output Offset parameters.



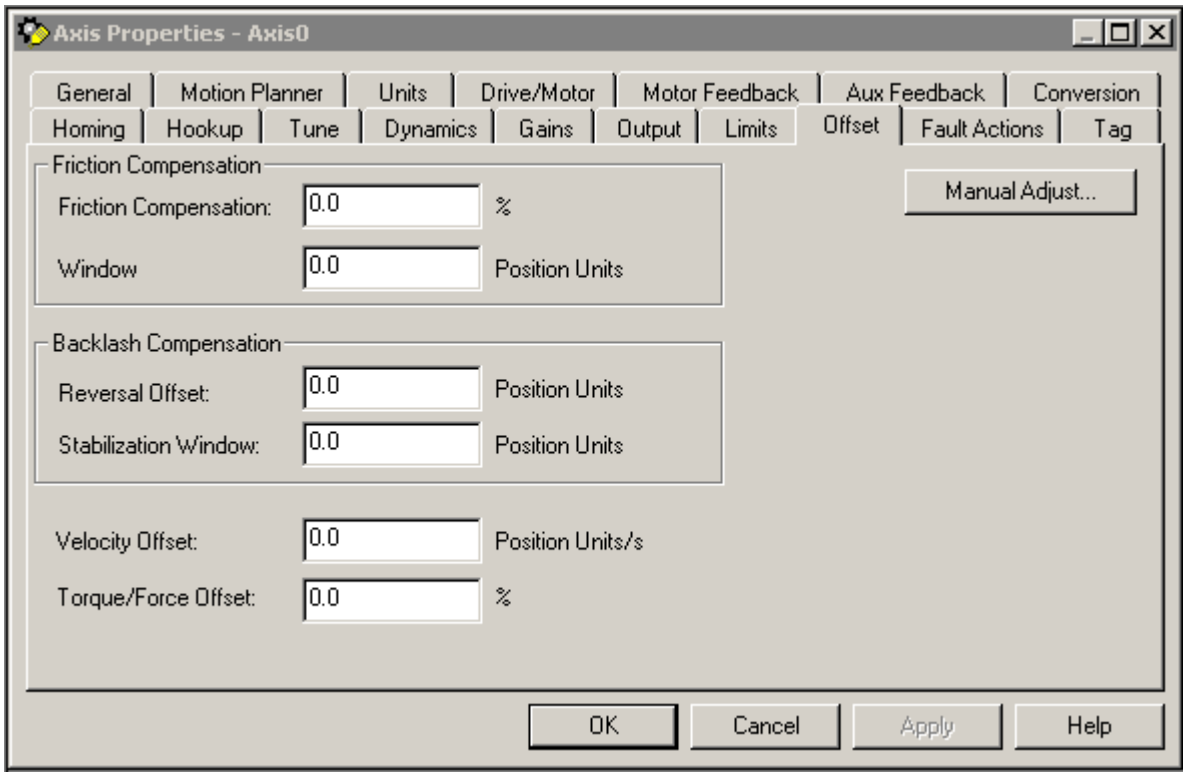
The Manual Adjust button is disabled when RSLogix 5000 software is in Wizard mode, and when offline edits to the above parameters have not yet been saved or applied.

Offset Tab - AXIS_SERVO_DRIVE

Use this tab to make offline adjustments to the following Servo Output values:

- Friction Compensation,
- Velocity Offset, and
- Torque Offset

for an axis of the type AXIS_SERVO_DRIVE configured as a Servo drive in the General tab of this dialog.



The parameters on this tab can be edited in either of two ways:

- edit on this tab by typing your parameter changes and then clicking on OK or Apply to save your edits
- edit in the Manual Adjust dialog: click on the Manual Adjust button to open the Manual Adjust dialog to this tab and use the spin controls to edit parameter settings. Your changes are saved the moment a spin control changes any parameter value.

The parameters on this tab become read-only and cannot be edited when the controller is online if the controller is set to Hard Run mode, or if a Feedback On condition exists.

When RSLogix 5000 software is offline, the following parameters can be edited and the program saved to disk using either the Save command or by clicking on the Apply button. You must re-download the edited program to the controller before it can be run.

Friction Compensation

The percentage of output level added to a positive current Servo Output value, or subtracted from a negative current Servo Output value, for the purpose of moving an axis that is stuck in place due to static friction.

It is not unusual for an axis to have enough static friction – called sticktion – that, even with a significant position error, the axis refuses to budge. Friction Compensation is used to break sticktion in the presence of a nonzero position error. This is done by adding, or subtracting, a percentage output level), called Friction Compensation to the Servo Output value.

The Friction Compensation value should be just less than the value that would break the "sticktion". A larger value can cause the axis to "dither", that is, move rapidly back and forth about the commanded position.

Friction Compensation Window

To address the issue of dither when applying Friction Compensation and hunting from the integral gain, a Friction Compensation Window is applied around the current command position when the axis is not being commanded to move. If the actual position is within the Friction Compensation Window the Friction Compensation value is applied to the Servo Output but scaled by the ratio of the position error to the Friction Compensation Window. Within the window, the servo integrators are also disabled. Thus, once the position error reaches or exceeds the value of the Friction Compensation Window attribute, the full Friction Compensation value is applied. If the Friction Compensation Window is set to zero, this feature is effectively disabled.

A nonzero Friction Compensation Window has the effect of softening the Friction Compensation as its applied to the Servo Output and reducing the dithering effect that it can create. This generally allows higher values of Friction Compensation to be applied. Hunting is also eliminated at the cost of a small steady-state error.

Backlash Compensation

Reversal Offset

Backlash Reversal Offset provides the capability to compensate for positional inaccuracy introduced by mechanical backlash. For example, power-train type applications require a high level of accuracy and repeatability during machining operations. Axis motion is often generated by a number of mechanical components, a motor, a gearbox, and a ball-screw that may introduce inaccuracies and that are subject to wear over their lifetime. Therefore, when an axis is commanded to reverse direction, mechanical play in the machine (through the gearing, ball-screw, and so on) may result in a small amount of motor motion without axis motion. As a result, the feedback device may indicate movement even though the axis has not physically moved.

If a value of zero is applied to the Backlash Reversal Offset, the feature is effectively disabled. Once enabled by a nonzero value, and the load is engaged by a reversal of the commanded motion, changing the Backlash Reversal

Offset can cause the axis to shift as the offset correction is applied to the command position.

Stabilization Window

The Backlash Stabilization Window controls the Backlash Stabilization feature in the servo control loop.

Properly configured with a suitable value for the Backlash Stabilization Window, entirely eliminates the gearbox buzz without sacrificing any servo performance. In general, this value should be set to the measured backlash distance. A Backlash Stabilization Window value of zero effectively disables the feature.

Velocity Offset

Provides a dynamic velocity correction to the output of the position servo loop, in position units per second.

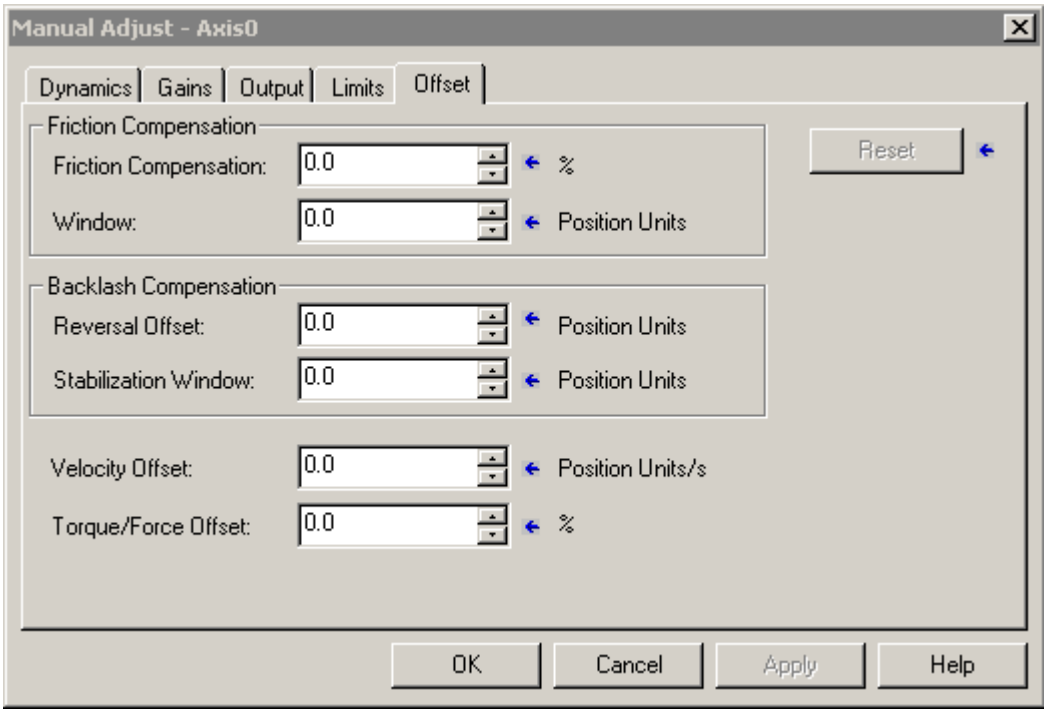
Torque/Force Offset

Provides a dynamic torque command correction to the output of the velocity servo loop, as a percentage of velocity servo loop output.

Manual Adjust

Click on this button to open the Offset tab of the Manual Adjust dialog for online editing of the Friction/Deadband Compensation, Backlash

Compensation, Velocity Offset, Torque Offset, and Output Offset parameters.

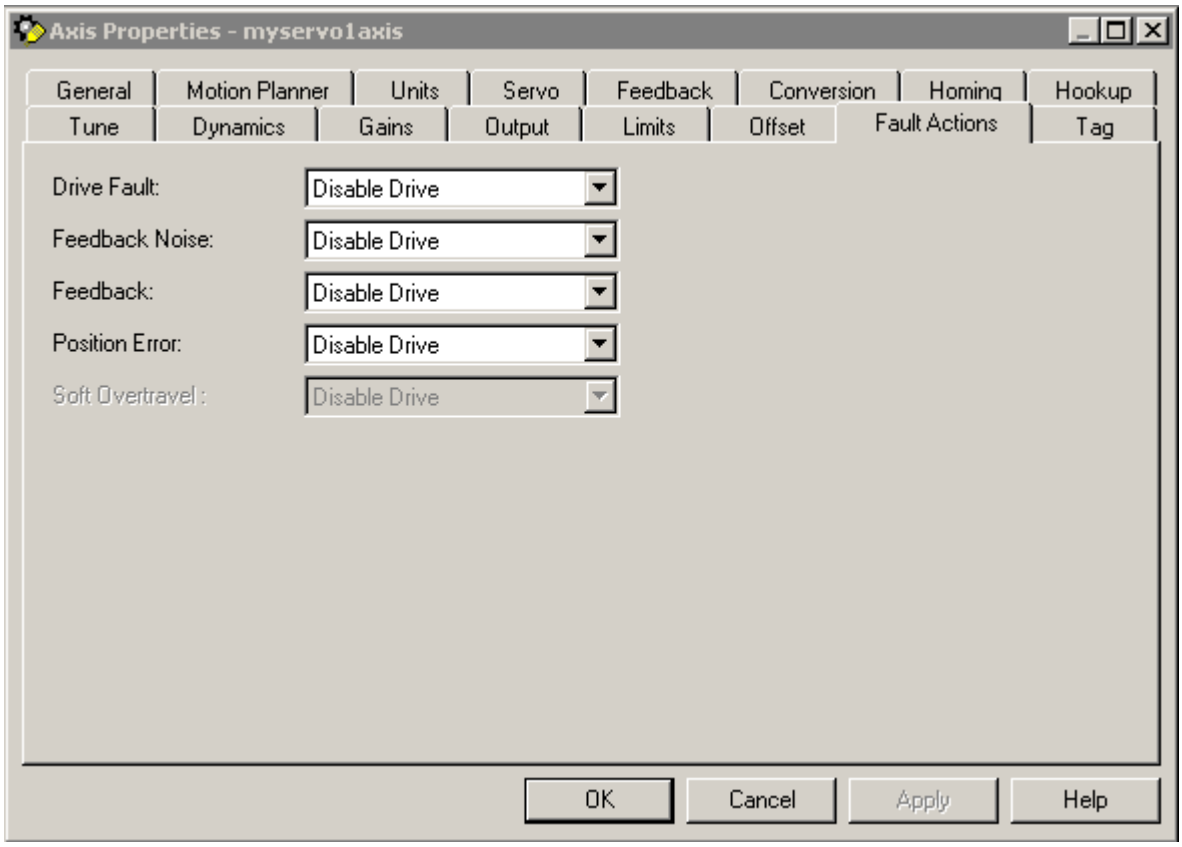


The Manual Adjust button is disabled when RSLogix 5000 software is in Wizard mode, and when offline edits to the above parameters have not yet been saved or applied.

Fault Actions Tab - AXIS_SERVO

Use this tab to specify the actions that are taken in response to these faults:

- Drive Fault
- Feedback Noise Fault
- Feedback Loss Fault
- Position Error Fault
- Soft Overtravel Fault



When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

When multiple workstations connect to the same controller using RSLogix 5000 software and invoke the Axis Wizard or Axis Properties dialog, the firmware allows only the first workstation to make any changes to axis attributes. The second workstation switches to a Read Only mode, indicated in the title bar, so that you may view the changes from that workstation, but not edit them.

Select one of the following fault actions for each fault type:

- Shutdown - If a fault action is set to Shutdown, then when the associated fault occurs, axis servo action is immediately disabled, the servo amplifier output is zeroed, and the appropriate drive enable output is deactivated. Shutdown is the most severe action to a fault and it is usually reserved for faults that could endanger the machine or the operator if power is not removed as quickly and completely as possible.

- Disable Drive - If a fault action is set to Disable Drive, then when the associated fault occurs, axis servo action is immediately disabled, the servo amplifier output is zeroed, and the appropriate drive enable output is deactivated.

- Stop Motion - If a fault action is set to Stop Motion, then when the associated fault occurs, the axis immediately starts decelerating the axis command position to a stop at the configured Maximum Deceleration Rate without disabling servo action or the servo modules Drive Enable output. This is the gentlest stopping mechanism in response to a fault. It is usually used for less severe faults. After the stop command fault action has stopped the axis, no further motion can be generated until the fault is first cleared.

- Status Only - If a fault action is set to Status Only, then when the associated fault occurs, no action is taken. The application program must handle any motion faults. In general, this setting should only be used in applications where the standard fault actions are not appropriate.

ATTENTION

Selecting the wrong fault action for your application can cause a dangerous condition resulting in unexpected motion, damage to the equipment, and physical injury or death. Keep clear of moving machinery.

Drive Fault

The Drive Fault field lets you specify the fault action to be taken when a drive fault condition is detected, for an axis with the Drive Fault Input enabled (in the Servo tab of this dialog) that is configured as Servo (in the General tab of this dialog). The available actions for this fault are Shutdown and Disable Drive.

Feedback Noise

The Feedback noise field lets you specify the fault action to be taken when excessive feedback noise is detected. The available actions for this fault are Shutdown, Disable Drive, Stop Motion, and Status Only.

Feedback Loss

The Feedback Loss field lets you specify the fault action to be taken when feedback loss condition is detected. The available actions for this fault are Shutdown, Disable Drive, Stop Motion, and Status Only.

Position Error

The Position Error field lets you specify the fault action to be taken when position error exceeds the position tolerance set for the axis, for an axis

configured as Servo (in the General tab of this dialog). The available actions for this fault are Shutdown, Disable Drive, Stop Motion, and Status Only.

Soft Overtravel

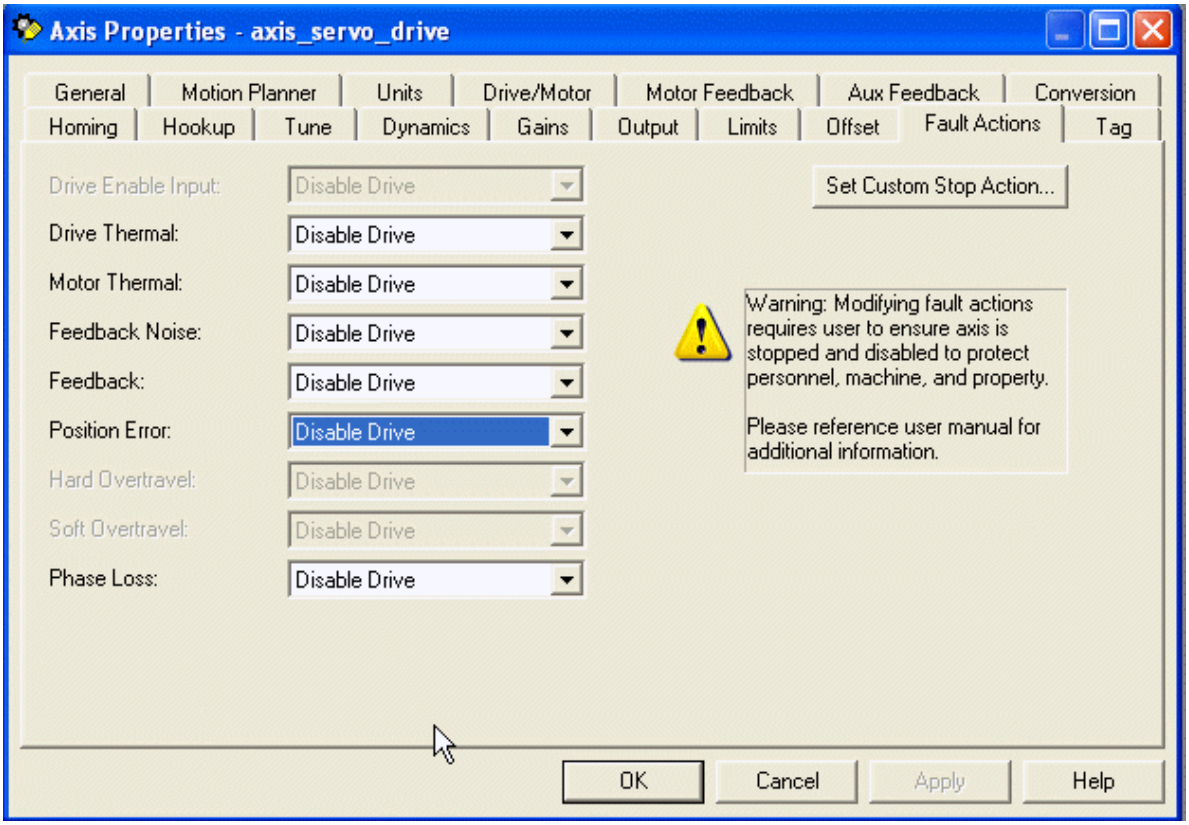
The Soft Overtravel field lets you specify the fault action to be taken when a software overtravel error occurs, for an axis with Soft Travel Limits enabled and configured (in the Limits tab of this dialog) that is configured as Servo (in the General tab of this dialog). The available actions for this fault are Shutdown, Disable Drive, Stop Motion, and Status Only.

Fault Actions Tab - AXIS_SERVO_DRIVE

Use this tab to specify the actions that are taken in response to the following faults:

- Drive Thermal Fault
- Motor Thermal Fault
- Feedback Noise Fault
- Feedback Fault
- Position Error Fault
- Hard Overtravel Fault
- Soft Overtravel Fault
- Phase Loss

for an axis of the type AXIS_SERVO_DRIVE.



When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

When multiple workstations connect to the same controller using RSLogix 5000 software and invoke the Axis Wizard or Axis Properties dialog, the firmware allows only the first workstation to make any changes to axis attributes. The second workstation switches to a Read Only mode, indicated in the title bar, so that you may view the changes from that workstation, but not edit them.

Select one of the following fault actions for each fault type:

- Shutdown - If a fault action is set to Shutdown, then when the associated fault occurs, axis servo action is immediately disabled, the servo amplifier output is zeroed, and the appropriate drive enable output is deactivated. Shutdown is the most severe action to a fault and it is usually reserved for faults that could endanger the machine or the operator if power is not removed as quickly and completely as possible.

- Disable Drive - If a fault action is set to Disable Drive, then when the associated fault occurs, it brings the axis to a stop by applying the Stopping Torque for up to the Stopping Time Limit. During this period the servo is active but no longer tracking the command reference from logix. Once the axis is stopped (or the stopping limit is exceeded) the servo and power structure are disabled.

- Stop Motion - If a fault action is set to Stop Motion, then when the associated fault occurs, the axis immediately starts decelerating the axis command position to a stop at the configured Maximum Deceleration Rate without disabling servo action or the servo modules Drive Enable output. This is the gentlest stopping mechanism in response to a fault. It is usually used for less severe faults. After the stop command fault action has stopped the axis, no further motion can be generated until the fault is first cleared.

- Status Only - If a fault action is set to Status Only, then when the associated fault occurs, no action is taken. The application program must handle any motion faults. In general, this setting should only be used in applications where the standard fault actions are not appropriate.

ATTENTION

Selecting the wrong fault action for your application can cause a dangerous condition. Keep clear of moving machinery.

Drive Thermal

Specifies the fault action to be taken when a Drive Thermal Fault is detected, for an axis configured as Servo (in the General tab of this dialog). The available actions for this fault are Shutdown, Disable Drive, Stop Motion, and Status Only.

Motor Thermal

Specifies the fault action to be taken when a Motor Thermal Fault is detected, for an axis configured as Servo (in the General tab of this dialog). The available actions for this fault are Shutdown, Disable Drive, Stop Motion, and Status Only.

Feedback Noise

Specifies the fault action to be taken when excessive feedback noise is detected. The available actions for this fault are Shutdown, Disable Drive, Stop Motion, and Status Only.

Feedback Specifies the fault action to be taken when Feedback Fault is detected. The available actions for this fault are Shutdown, Disable Drive, Stop Motion, and Status Only.

Position Error Specifies the fault action to be taken when position error exceeds the position tolerance set for the axis, for an axis configured as Servo (in the General tab of this dialog). The available actions for this fault are Shutdown, Disable Drive, Stop Motion, and Status Only.

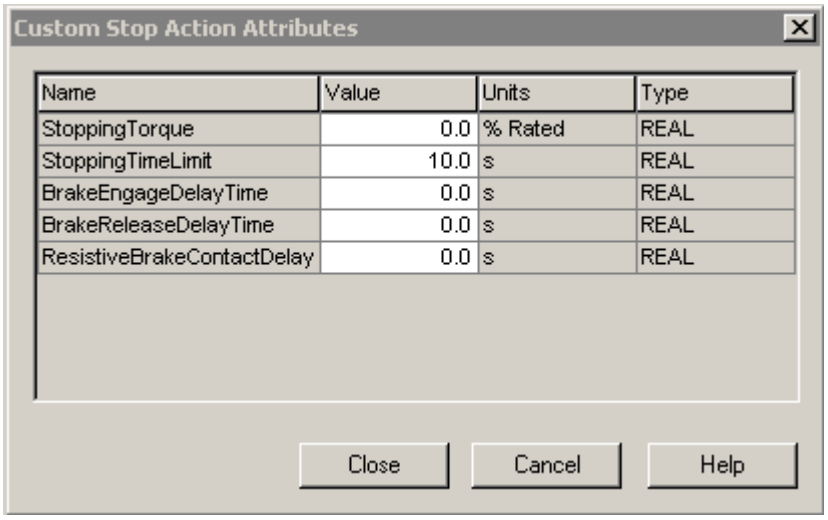
Hard Overtravel Specifies the fault action to be taken when an axis encounters a travel limit switch, for an axis configured as Servo (in the General tab of this dialog). The available actions for this fault are Shutdown, Disable Drive, Stop Motion, and Status Only.

Soft Overtravel Specifies the fault action to be taken when a software overtravel error occurs, for an axis with Soft Travel Limits enabled and configured (in the Limits tab of this dialog) that is configured as Servo (in the General tab of this dialog). The available actions for this fault are Shutdown, Disable Drive, Stop Motion, and Status Only.

Phase Loss Specifies the fault action to be taken when a phase loss situation occurs for an axis configured as Servo (on the General tab of this dialog). The available actions for this fault are Shutdown, Disable Drive, Stop Motion and Status Only. The default is Shutdown. When Status Only is chosen Logix 5000 motion commands continue and the drive uses available stored DC bus energy to operate the axes.

Set Custom Stop Action

Opens the Custom Stop Action Attributes dialog.



Use this dialog to monitor and edit the Stop Action-related attributes.

When a parameter transitions to a read-only state, any pending changes to parameter values are lost, and the parameter reverts to the most recently saved parameter value.

When multiple workstations connect to the same controller using RSLogix 5000 software and invoke the Axis Wizard or Axis Properties dialog, the firmware allows only the first workstation to make any changes to axis attributes. The second workstation switches to a Read Only mode, indicated in the title bar, so that you may view the changes from that workstation, but not edit them.

Attributes

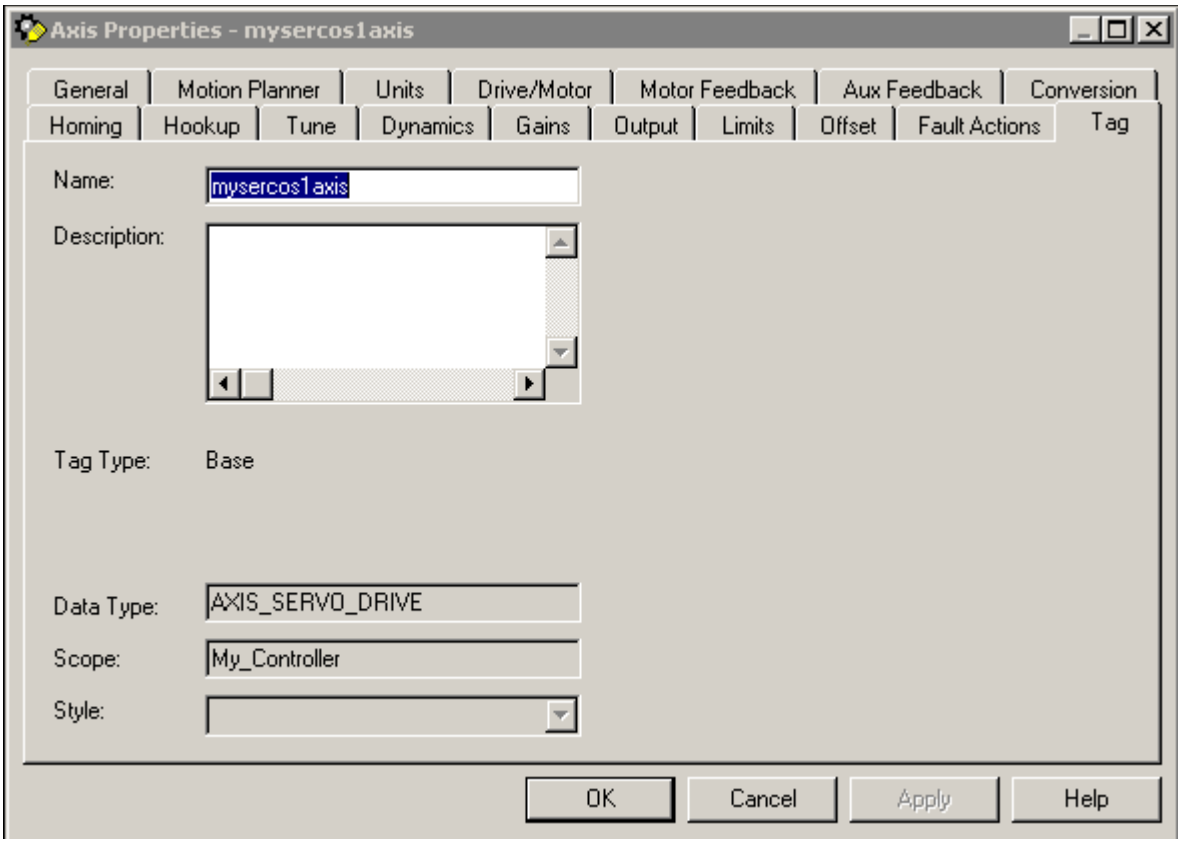
The following attribute, or parameter, values can be monitored and edited in this dialog box.

Attribute	Description
StoppingTorque	This attribute displays the amount of torque available to stop the motor. This attribute has a value range of 0...1000.
StoppingTimeLimit	This attribute displays the maximum amount of time that the drive amplifier remains enabled while trying to stop. It is useful for very slow velocity rate change settings. This attribute has a value range of 0...6553.5.
BrakeEngageDelayTime	When servo axis is disabled and the drive decelerates to a minimum speed, the drive maintains torque until this time has elapsed. This time allows the motor's brake to be set. This attribute has a value range of 0...6.5535.
BrakeReleaseDelayTime	When the servo axis is enabled, the drive activates the torque to the motor but ignores the command values from the Logix controller until this time has elapsed. This time allows the motor's brake to release. This attribute has a value of 0...6.5535.
ResistiveBrakeContactDelay	The Resistive Brake Contact Delay attribute is used to control an optional external Resistive Brake Module (RBM). The RBM sits between the drive and the motor and uses an internal contactor to switch the motor between the drive and a resisted load.

Tag Tab

Use this tab to modify the name and description of the axis. When you are online, all of the parameters on this tab transition to a read-only state, and

cannot be modified. If you go online before you save your changes, all pending changes revert to their previously-saved state.



Name Displays the name of the current tag. You can rename this tag, if you wish.

Description Displays the description of the current tag, if any is available. You can edit this description, if you wish.

Tag Type Indicates the type of the current tag. This type may be:

- Base
- Alias
- Consumed

Displays the data type associated with the current tag.

Data Type Displays the axis data type of the current tag.

Scope Displays the scope of the current tag. The scope is either controller scope, or program scope, based on one of the existing programs in the controller.

Style Displays the default style in which to display the value of the tag. Note that style is only applicable to an atomic tag; a structure tag does not have a display style.

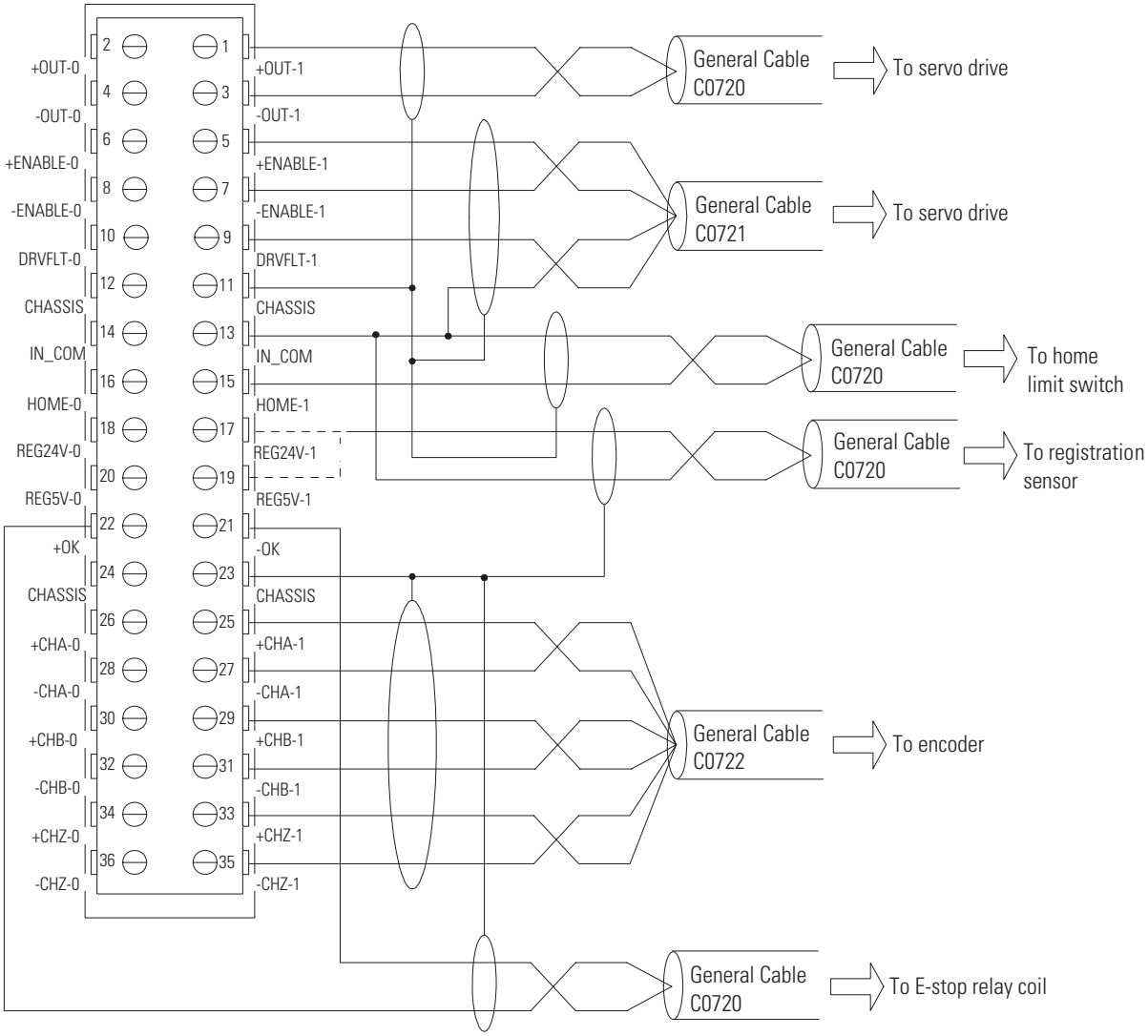
Wiring Diagrams

Introduction

Use the diagrams in this appendix to wire the motion control equipment of your control system.

To wire this	See page
1756-M02AE Module	258
Ultra 100 Series Drive	259
Ultra 200 Series Drive	259
Ultra3000 Drive	261
1394 Servo Drive (in Torque Mode only)	263
1756-M02AS Module	265
1756-HYD02 Application Example	266
1756-HYD02 Module	267
LDTs	268
Temposonic GH Feedback Device	269
24V Registration Sensor	270
5V Registration Sensor	270
Home Limit Switch Input	271
OK Contacts	271

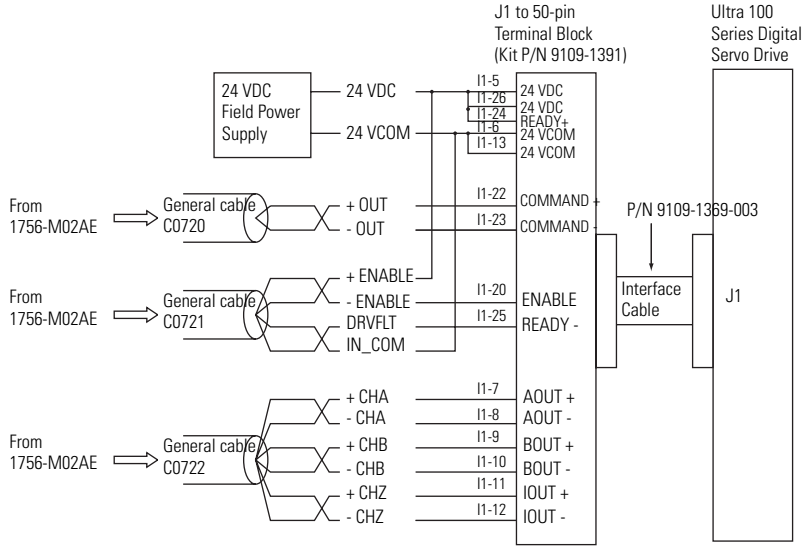
1756-M02AE Module



Notes

This example shows the wiring for Axis 1 Wire Axis 0 the same way.

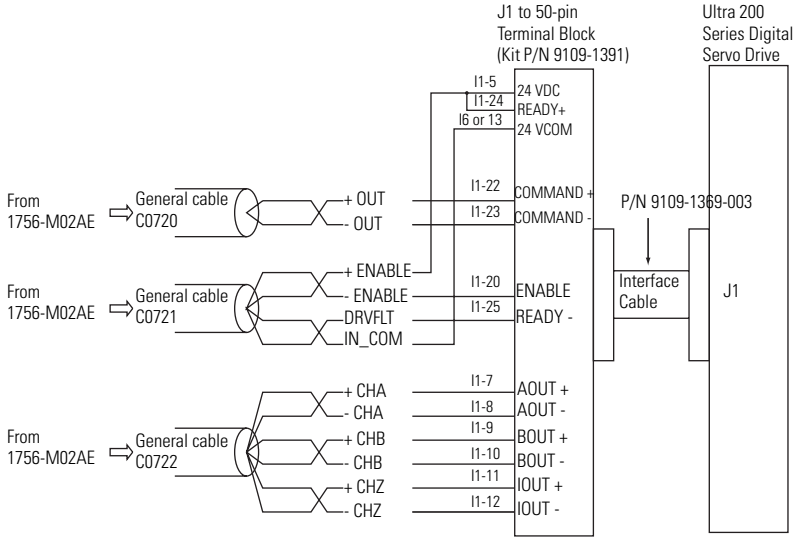
Ultra 100 Series Drive



Notes

- This is an example of one way to wire the drive.
- See Ultra 100 Series Drive Installation Manual, publication number 1398-5.2, for other configurations.

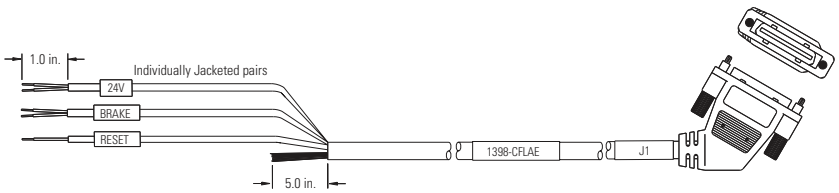
Ultra 200 Series Drive



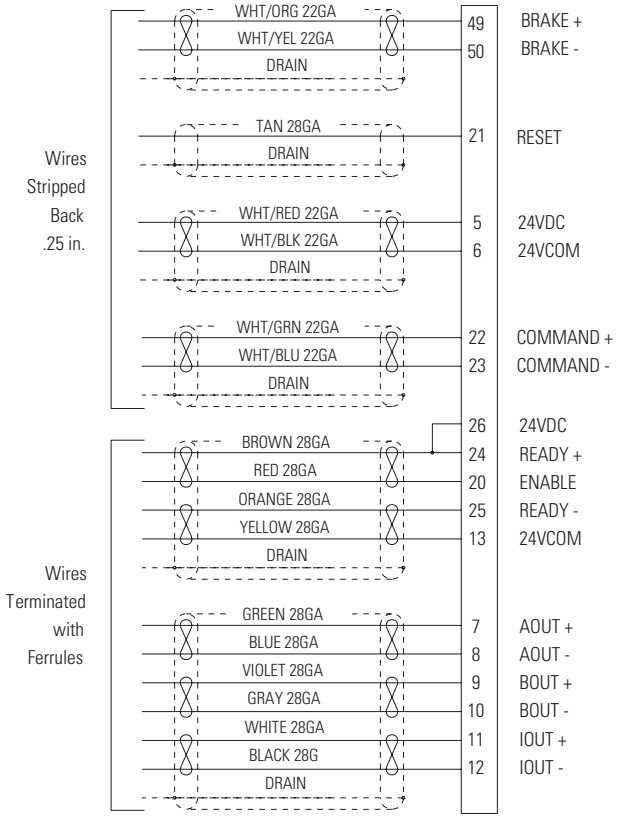
Notes

- This is an example of one way to wire the drive.
- See Ultra 200 Series Drive Installation Manual, publication number 1398-5.0, for other configurations.

1398-CFLAExx Cable

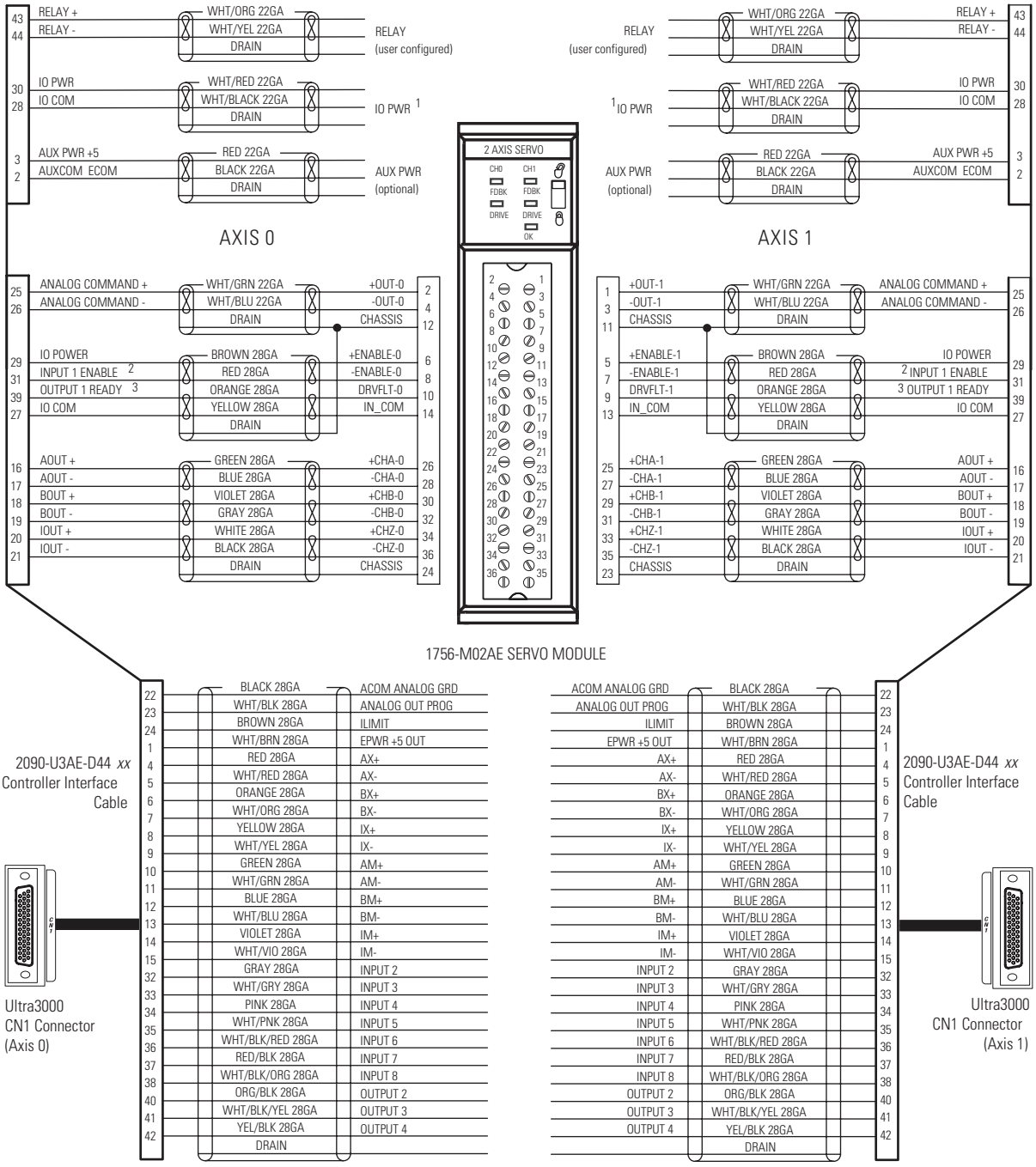


Pinouts for 1398-CFLAExx Cable



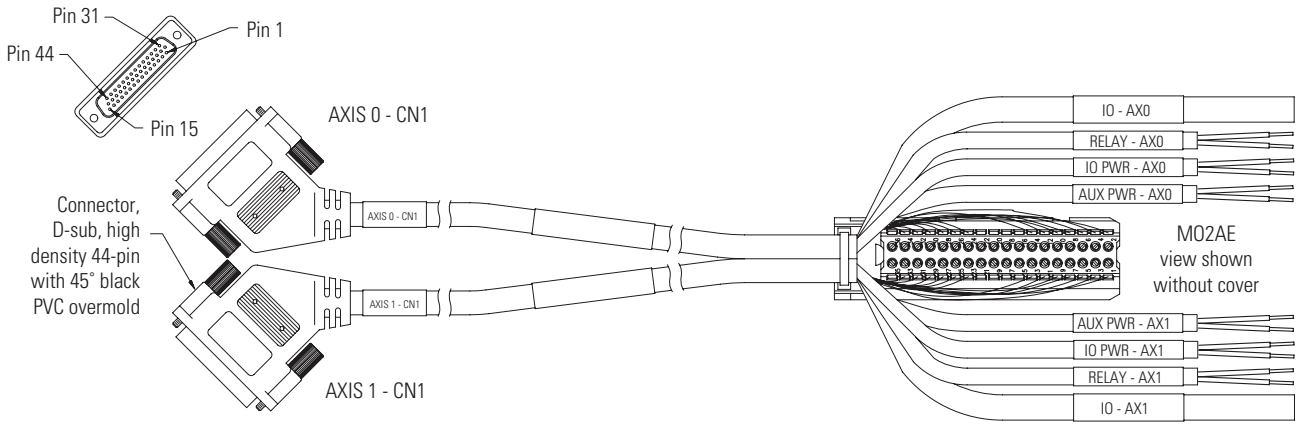
Ultra3000 Drive

Ultra3000 to 1756-M02AE Interconnect diagram.

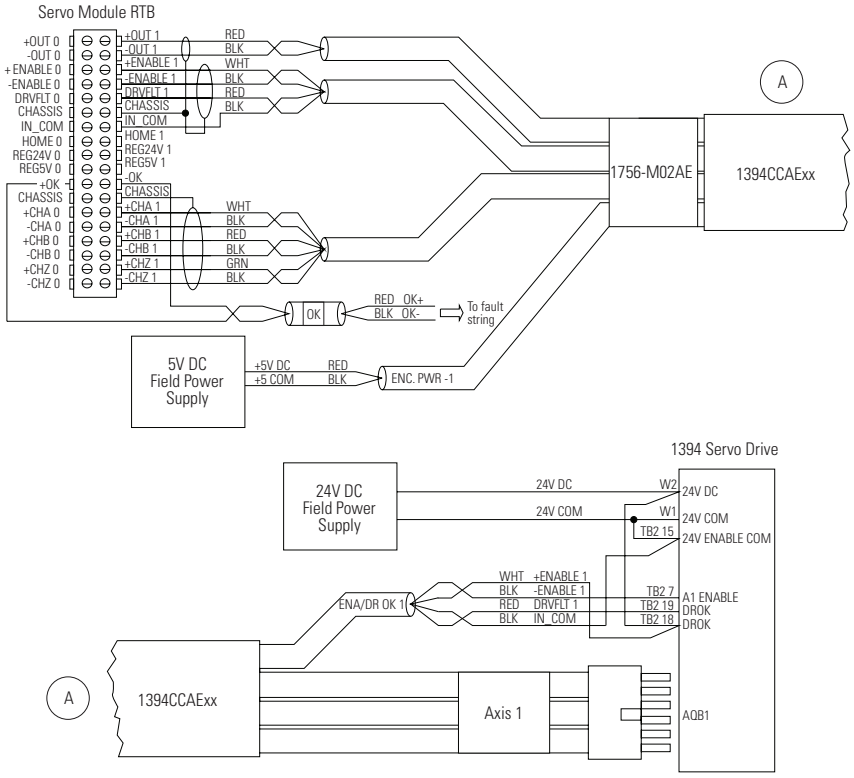


For more information, see Ultra3000 Digital Servo Drives Installation Manual, publication number 2098-IN003.

2090-U3AE-D44*xx* Cable.



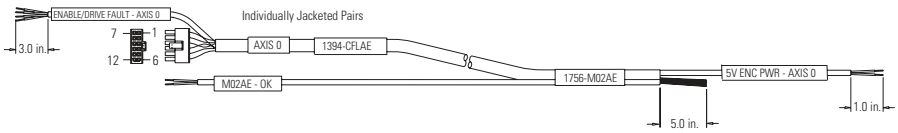
1394 Servo Drive (in Torque Mode only)



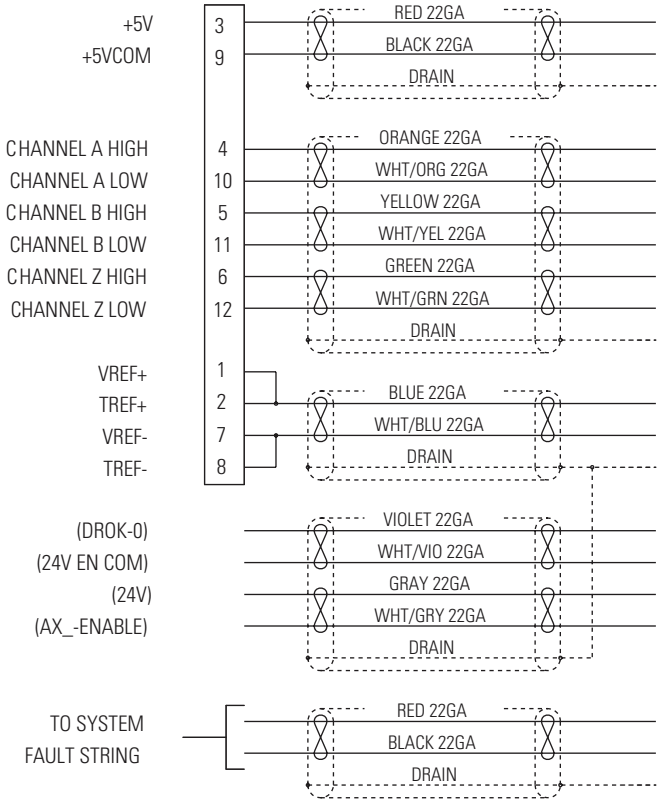
Notes

- The wiring diagram illustrates Axis 1 wiring only. Other configurations are possible.
- The 1394CCAE*xx* cable is wired to connect to torque command reference input pins.
- The *xx* in the cable number is the length of the cable.
- An external +5V power supply is required to power the encoder driver circuit of the 1394 servo drive. Because this connection is shared by all four axis encoder driver circuits, only one connection is needed to the +5V field supply.

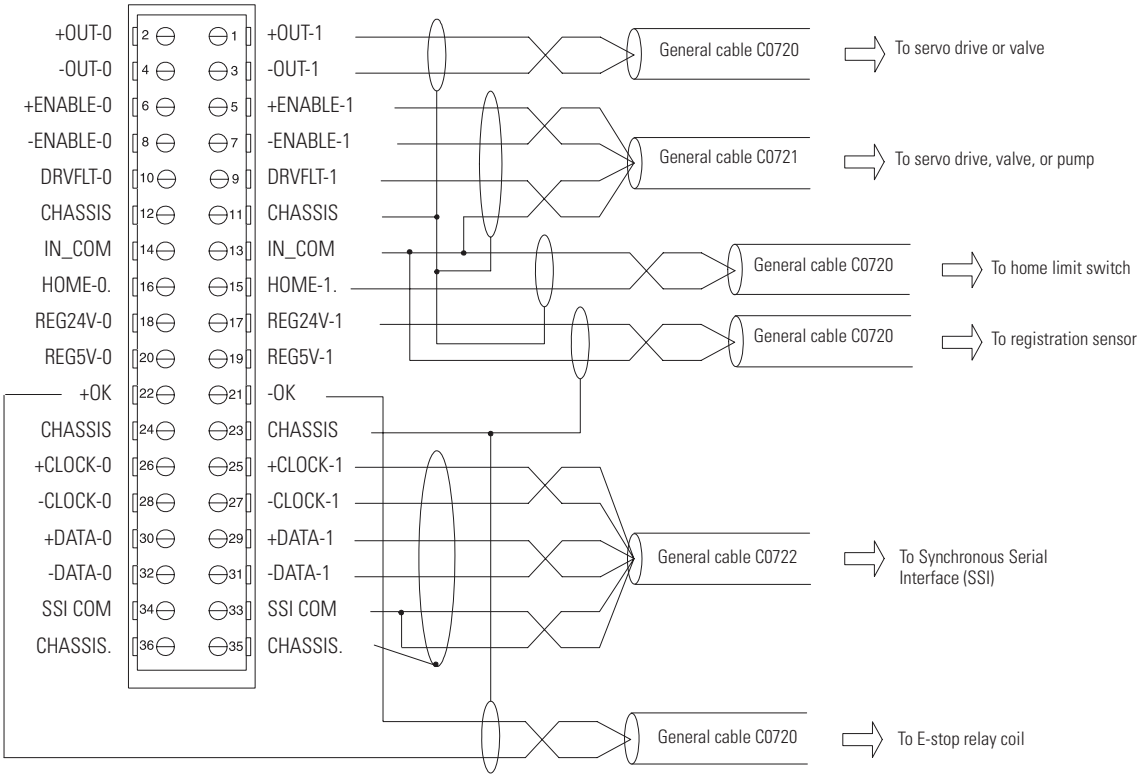
1394-CFLAE*xx* Cable



Pinouts for the 1394-CFLAE



1756-M02AS Module

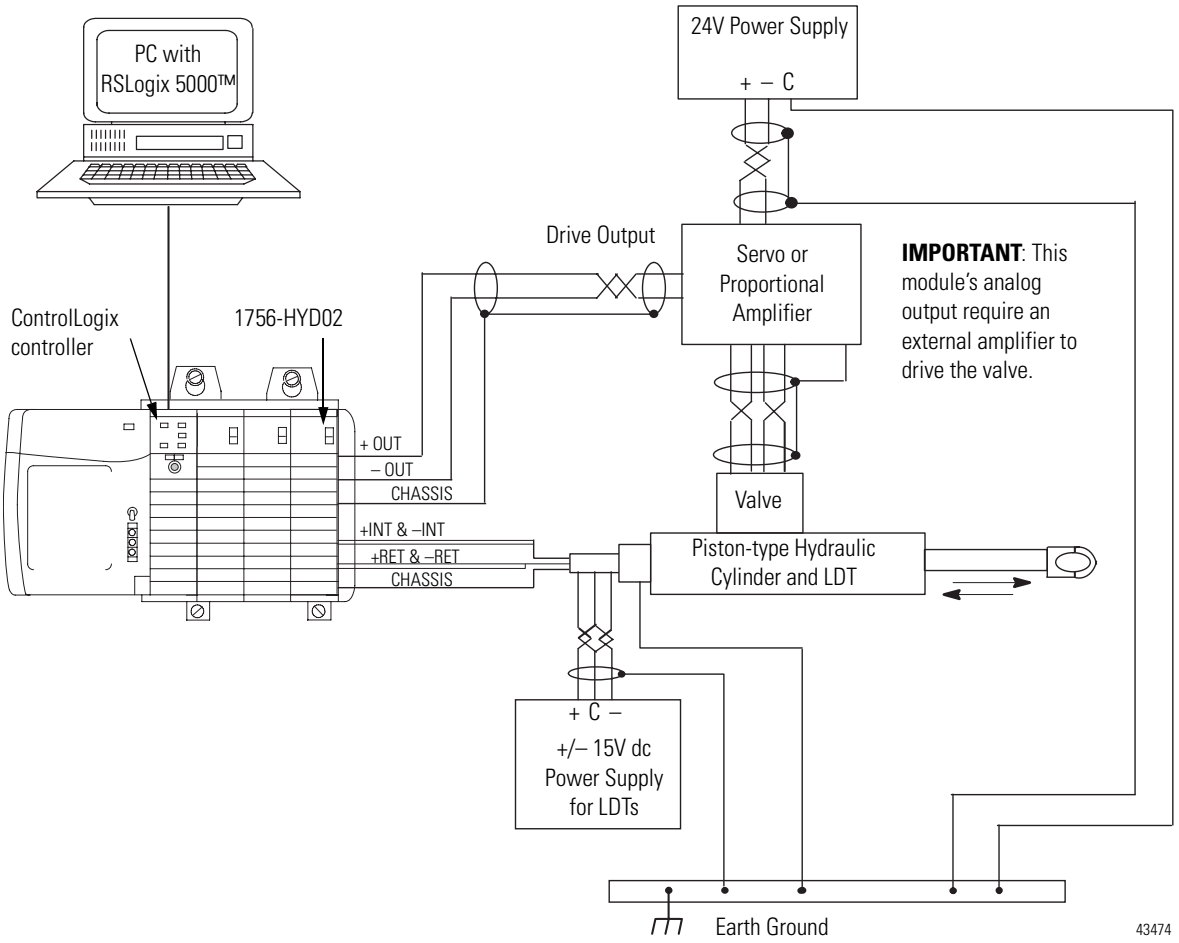


Notes

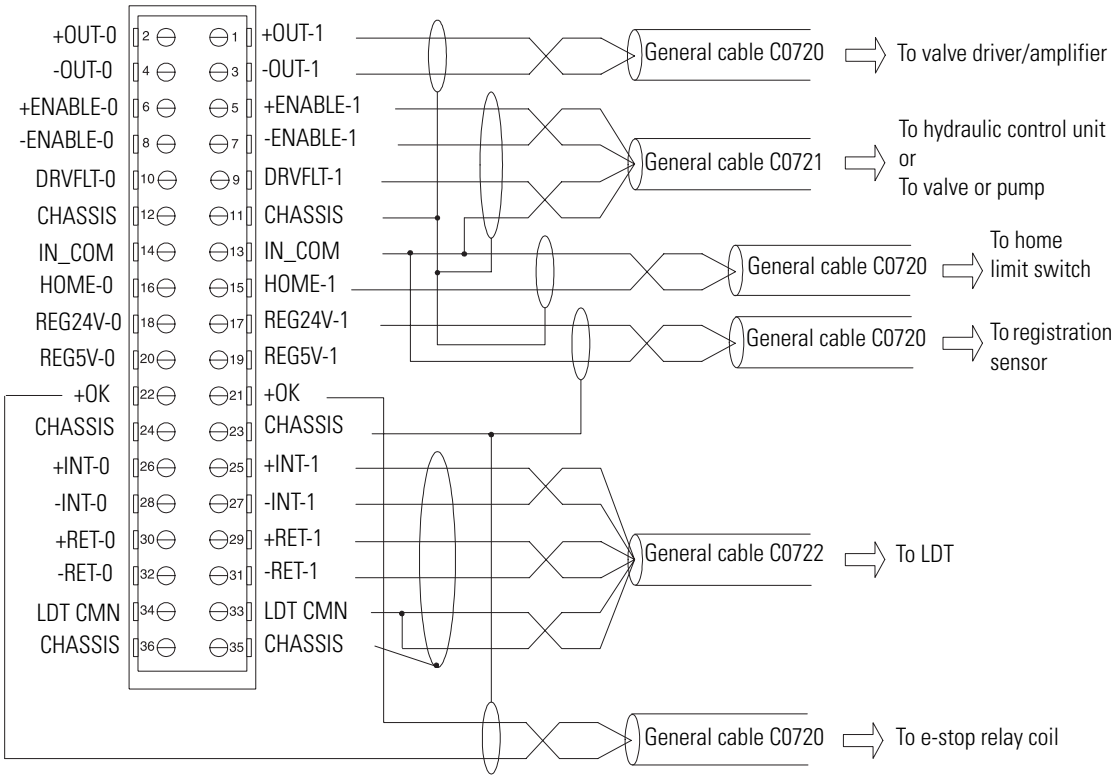
This example shows the wiring for Axis 1 Wire Axis 0 the same way.

1756-HYD02 Application Example

This example uses a 1-axis loop with a differential LDT input.



1756-HYD02 Module



Notes

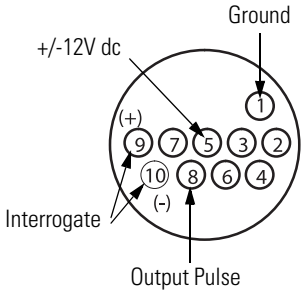
- This example shows the wiring for Axis 1. Wire Axis 0 the same way.
- Use transducers that use an external interrogation signal.
- Do not exceed the specified isolation voltage between power sources.

LDTs

These diagrams show the connections for Temposonic and Balluff LDTs.

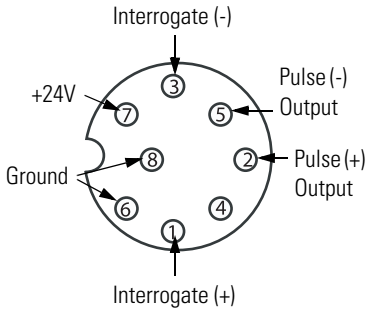
IMPORTANT Other suppliers also have compatible LDTs. Before you connect an LDT to your module, make sure that it is the best LDT for your application.

Temposonics II, RPM or DPM

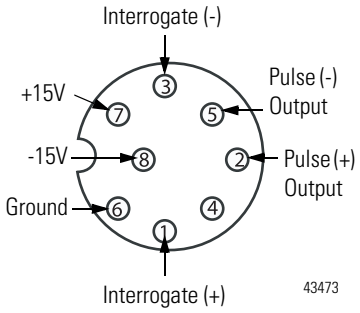


Balluff BTL type

24V Connections



+/- 15V Connections



No shield connections on these examples

This table lists the LDT connections.

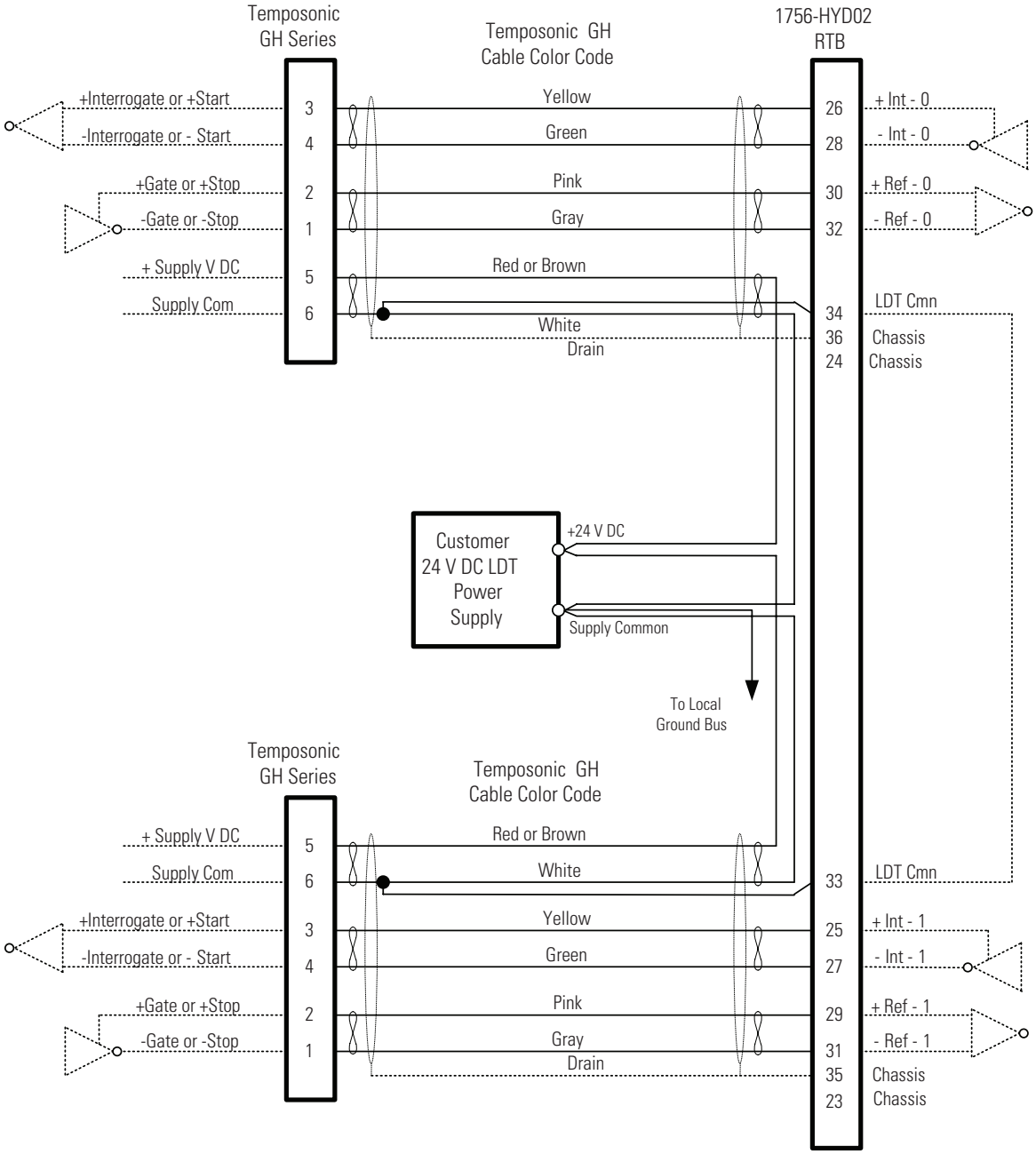
LDT Connections for Fabricating Your Own LDT Cable

Function[1]	1756-HYD02 RTB Wiring (Numbers below represent terminal numbers)		Temposonics II[2] RPM or DPM	Balluff BTL type	
	Channel 0	Channel 1		24V dc	+/- 15V dc
(+) Interrogate	26	25	9 - Yellow	1 - Yellow	1 - Yellow
(-) Interrogate	28	27	10 - Green	3 - Pink	3 - Pink
Power Supply	N/A		5 - Red (+/-12V)	7 - Brown (+24V)	7 - Brown (+15V) 8 - White (-15V)
Ground	34	33	1 - White	6 - Blue 8 - White	6 - Blue
Output Pulse	30 (+) 32 (-)	29 (+) 31 (-)	8 - Purple	2 - Gray (+) 5 - Green (-)	2 - Gray (+) 5 - Green (-)

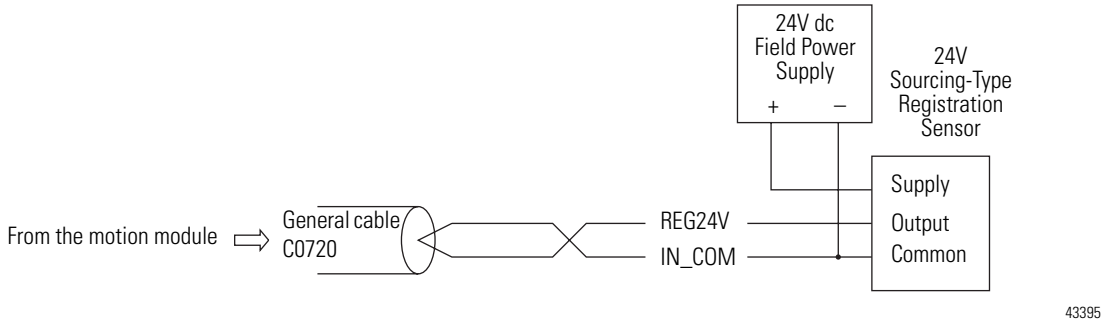
(1) (+) and (-) wires of the same function should be a twisted pair within the cable.

(2) Do not connect to pins 2, 3, 4, 6 or 7

Temposonic GH Feedback Device



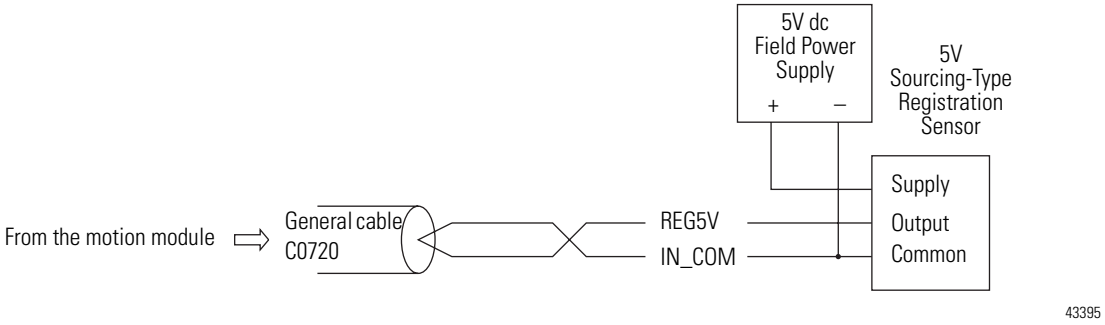
24V Registration Sensor



Notes

- Use sourcing-type registration sensors.
- Wire the inputs so that they get source current from the sensor.
- Don't use current sinking sensor configurations because the registration input common (IN_ COM) is shared with the other 24V servo module inputs.

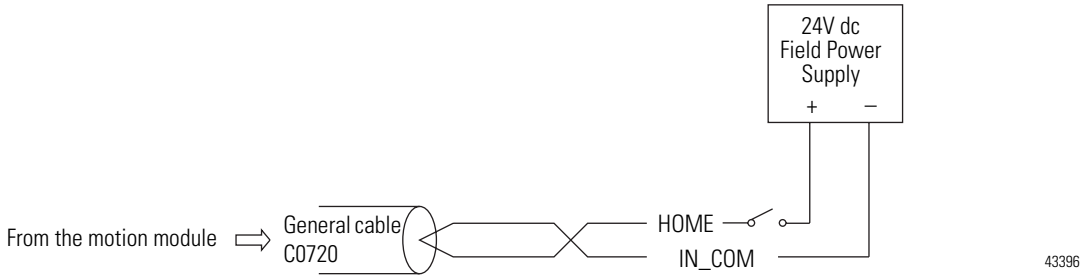
5V Registration Sensor



Notes

- Use sourcing-type registration sensors.
- Wire the inputs so that they get source current from the sensor.
- Don't use current sinking sensor configurations because the registration input common (IN_ COM) is shared with the other 24V servo module inputs.

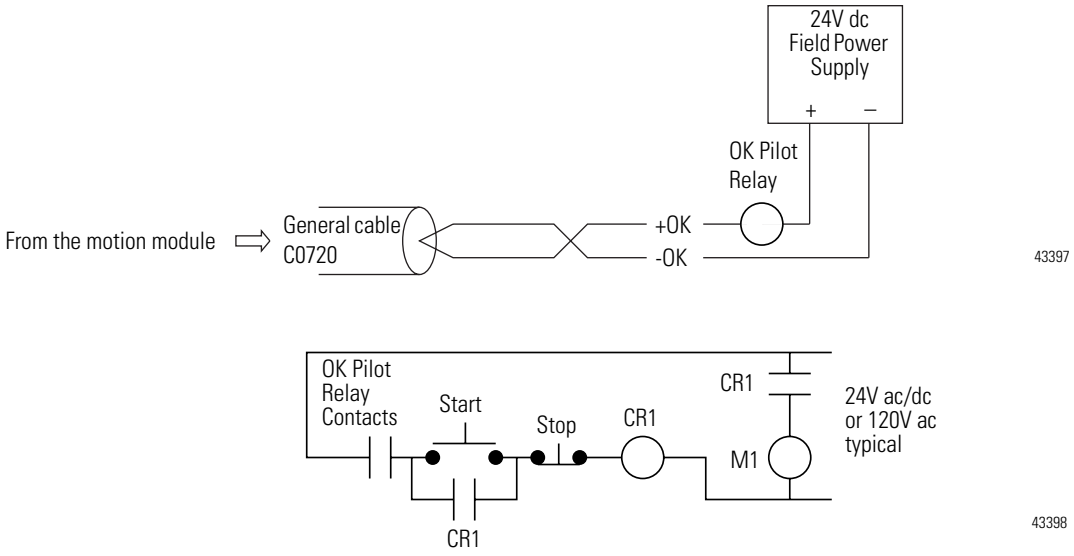
Home Limit Switch Input



Notes

- The home limit switch inputs to the servo module are designed for 24V dc nominal operation.
- Wire these inputs for current sourcing operation.

OK Contacts



Notes

- Use the OK relay contacts to connect to an E-stop string that controls power to the associated pumps or drives.
- The OK contacts are rated to drive an external 24V dc pilot relay (for example, Allen-Bradley 700-HA32Z24) whose contacts can be incorporated into the E-stop string.

Notes:

Axis Attributes

Introduction

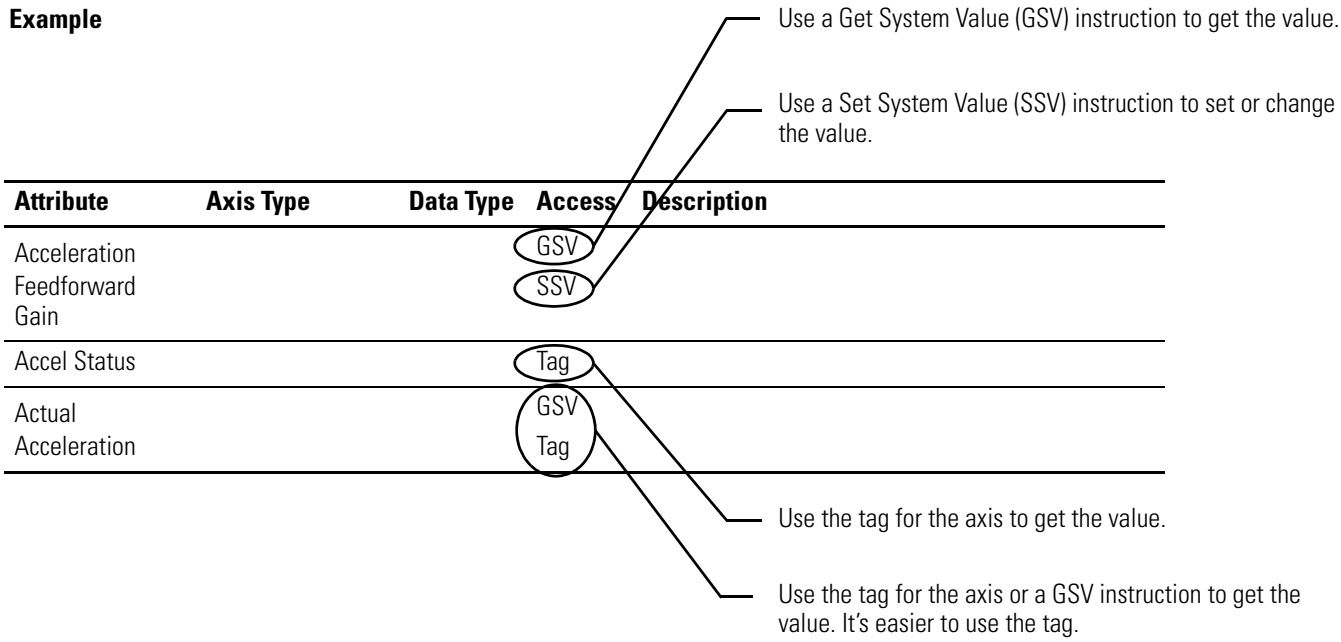
Use this chapter to get configuration, status, and fault information about an axis. The controller stores information about an axis as attributes of the axis.

Topic	Page
How to Access Attributes	273
Axis Attributes	274

How to Access Attributes

The Access column shows how to access the attribute.

Example


Use a Get System Value (GSV) instruction to get the value.
Use a Set System Value (SSV) instruction to set or change the value.
Use the tag for the axis to get the value.
Use the tag for the axis or a GSV instruction to get the value. It's easier to use the tag.

Attribute	Axis Type	Data Type	Access	Description
Acceleration Feedforward Gain			GSV SSV	
Accel Status			Tag	
Actual Acceleration			GSV Tag	

Axis Attributes

This table describes each attribute of an axis.

Attribute	Axis Type	Data Type	Access	Description
Absolute Feedback Enable	AXIS_SERVO	SINT	GSV SSV	**Important**: Use this attribute only for an axis of a 1756-HYD02 or 1756-M02AS module. This attribute controls whether or not the servo module uses the absolute position capability of the feedback device. If Absolute Feedback Enable is set to True, the servo module adds the Absolute Feedback Offset to the current position of the feedback device to establish the absolute machine reference position. Since absolute feedback devices retain their position reference even through a power-cycle, the machine reference system can be restored at power-up. To establish a suitable value for the Absolute Feedback Offset attribute the MAH instruction may be executed with the Home Mode configured for Absolute (the only valid option when Absolute Feedback Enable is True). When executed, the servo module will compute the Absolute Feedback Offset as the difference between the configured value for Home Position and the current absolute feedback position of the axis. The computed Absolute Feedback Offset is immediately applied to the axis upon completion of the MAH instruction. Because the actual position of the axis is re-referenced during execution of the MAH instruction, the servo loop must not be active. If the servo loop is active the MAH instruction errors. If Absolute Feedback Enable is set to False, the servo module ignores the Absolute Feedback Offset and treats the feedback device as an incremental position transducer. In this case, a homing or redefine position operation is therefore needed to establish the absolute machine reference position. The Absolute Home Mode in this case is considered invalid. This attribute is configurable if the Transducer Type is set to SSI. For an LDT transducer the Absolute Feedback Enable is forced to True. For an AQB transducer the Absolute Feedback Enable is forced to False.

Attribute	Axis Type	Data Type	Access	Description
Absolute Feedback Offset	AXIS_SERVO	REAL	GSV SSV	Position Units **Important** · Use this attribute only for an axis of a 1756-HYD02 or 1756-M02AS module. · Set the Absolute Feedback Enable attribute to True. This attribute is used to determine the relative distance between the absolute position of the feedback device and the absolute position of the machine. At power-up this attribute is sent to the servo module and added to the current position of the feedback device to restore the absolute machine position reference. If the axis is configured for Linear operation, absolute position may be recovered after power cycle as long as the feedback device has not exceeded its range limit. If the feedback device rolls over its count range, the absolute position of the axis is no longer valid. If the axis is configured for Rotary operation, the servo module is responsible for adjusting the Absolute Feedback Offset dynamically based on the configured Unwind value and the rollover of the absolute feedback device. If necessary, absolute position may be recovered after power cycle by periodically updating the controller's Absolute Feedback Offset value. This can be done by selecting the Absolute Feedback Offset enumeration for one of the Axis Info Select attributes.
Absolute Reference Status	AXIS_SERVO_DRIVE	BOOL	Tag	**If the bit is** / **Then** ON: An absolute homing procedure happend. The bit stays set until either of these happen: · The drive resets its configuration parameters to default values. · The axis does an active or passive home or redefine position. OFF: The position of the axis has not been, or is no longer, referenced to the absolute machine reference system established by an absolute homing procedure.
Accel Limit Status	AXIS_SERVO_DRIVE	BOOL	Tag	Set when the magnitude of the commanded acceleration to the velocity servo loop input is greater than the configured Velocity Limit.
Accel Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set if the axis is currently being commanded to accelerate. Use the Accel Status bit and the Decel Status bit to see if the axis is accelerating or decelerating. If both bits are off, then the axis is moving at a steady speed or is at rest.

Attribute	Axis Type	Data Type	Access	Description
Acceleration Command	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important**: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. Acceleration Command in Position Units / Sec2 Acceleration Command is the current acceleration reference to the output summing junction, in the configured axis Position Units per Second^2, for the specified axis. The Acceleration Command value, hence, represents the output of the inner velocity control loop. Acceleration Command is not to be confused with Command Velocity, which represents the rate of change of Command Position input to the position servo loop.
Acceleration Data Scaling	AXIS_SERVO_DRIVE	INT	GSV	This attribute is derived from the Drive Units attribute. See IDN 160 in IEC 1491.
Acceleration Data Scaling Exp	AXIS_SERVO_DRIVE	INT	GSV	This attribute is derived from the Drive Units attribute. See IDN 162 in IEC 1491.
Acceleration Data Scaling Factor	AXIS_SERVO_DRIVE	DINT	GSV	This attribute is derived from the Drive Units attribute. See IDN 161 in IEC 1491.
Acceleration Feedback	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important**: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. Acceleration Feedback in Position Units / Sec2 Acceleration Feedback is the actual velocity of the axis as estimated by the servo module, in the configured axis Position Units per Second^2. The Estimated Acceleration is calculated by taking the difference in the Estimated Velocity over the servo update interval. Acceleration Feedback is a signed value—the sign (+ or -) depends on which direction the axis is currently moving.

Attribute	Axis Type	Data Type	Access	Description
Acceleration Feedforward Gain	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	%

AXIS_SERVO

When you connect to a torque servo drive, use the Acceleration Feedforward Gain to give the Torque Command output necessary to generate the commanded acceleration. It does this by scaling the current Command Acceleration by the Acceleration Feedforward Gain and adding it as an offset to the Servo Output generated by the servo loop. With this done, the servo loops do not need to generate much of a contribution to the Servo Output, hence the Position and/or Velocity Error values are significantly reduced. Hence, when used in conjunction with the Velocity Feedforward Gain, the Acceleration Feedforward Gain lets the following error of the servo system during the acceleration and deceleration phases of motion be reduced to nearly zero. This is important in applications such as electronic gearing and synchronization where the actual axis position must not significantly lag behind the commanded position at any time.

When you connect to a velocity servo drive, use Acceleration Feedforward to add a term to the Velocity Command that is proportional to the commanded acceleration. This can be effective in cases where the external drive shows a steady-state velocity error during acceleration and deceleration.

The best value for Acceleration Feedforward depends on the drive configuration. Excessive Acceleration Feedforward values tend to produce axis overshoot. For torque servo drive applications the best value for Acceleration Feedforward is theoretically 100%. However, the value may need to be increased slightly to accommodate servo loops with non-infinite loop gain and other application considerations. For velocity servo drive applications the best value for Acceleration Feedforward is highly dependent on the drive's speed scaling and servo loop configuration. A value of 100%, in this case, means only that 100% of the commanded acceleration value is applied to the velocity command summing junction and may not be even close to the optimal value.

To find the best Acceleration Feedforward Gain, run a simple project that jogs the axis in the positive direction and monitors the Position Error of the axis during the jog. Usually Acceleration Feedforward is used in tandem with Velocity Feedforward to achieve near zero following error during the entire motion profile. To fine tune the Acceleration Feedforward Gain, the Velocity Feedforward Gain must first be optimized using the procedure described above. While capturing the peak Position Error during the acceleration phase of the jog profile, increase the Acceleration Feedforward Gain until the peak Position Error is as small as possible, but still positive. If the peak Position Error during the acceleration ramp is negative, the actual position of the axis is *ahead* of the command position during the acceleration ramp. If this occurs, decrease the Acceleration Feedforward Gain such that the Position Error is again positive. To be thorough the same procedure should be done for the deceleration ramp to verify that the peak Position Error during deceleration is acceptable. Note that reasonable maximum velocity, acceleration, and deceleration values must be entered to jog the axis.

Continued on next page

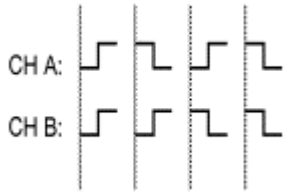
Attribute	Axis Type	Data Type	Access	Description
Acceleration Feedforward Gain (cont.)				**AXIS_SERVO_DRIVE** The Acceleration Feedforward Gain attribute is used to provide the Torque Command output necessary to generate the commanded acceleration. It does this by scaling the current Command Acceleration by the Acceleration Feedforward Gain and adding it as an offset to the Servo Output generated by the servo loop. With this done, the servo loops do not need to generate much control effort, hence the Position and/or Velocity Error values are significantly reduced. When used in conjunction with the Velocity Feedforward Gain, the Acceleration Feedforward Gain allows the following error of the servo system during the acceleration and deceleration phases of motion to be reduced to nearly zero. This is important in applications such as electronic gearing and synchronization applications where it is necessary that the actual axis position not significantly lag behind the commanded position at any time. The optimal value for Acceleration Feedforward is 100% theoretically. In reality, however, the value may need to be tweaked to accommodate torque loops with non-infinite loop gain and other application considerations. One thing that may force a smaller Acceleration Feedforward value is that increasing amounts of feedforward tends to exacerbate axis overshoot. When necessary, the Acceleration Feedforward Gain may be "tweaked" from the 100% value by running a simple user program that jogs the axis in the positive direction and monitors the Position Error of the axis during the jog. Usually Acceleration Feedforward is used in tandem with Velocity Feedforward to achieve near zero following error during the entire motion profile. To fine-tune the Acceleration Feedforward Gain, the Velocity Feedforward Gain must first be optimized using the procedure described above. While capturing the peak Position Error during the acceleration phase of the jog profile, increase the Acceleration Feedforward Gain until the peak Position Error is as small as possible, but still positive. If the peak Position Error during the acceleration ramp is negative, the actual position of the axis is *ahead* of the command position during the acceleration ramp. If this occurs, decrease the Acceleration Feedforward Gain such that the Position Error is again positive. To be thorough the same procedure should be done for the deceleration ramp to verify that the peak Position Error during deceleration is acceptable. Note that reasonable maximum velocity, acceleration, and deceleration values must be entered to jog the axis.
Acceleration Limit Bipolar	AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units / sec^2 This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually don't have to change it.
Acceleration Limit Negative	AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units / sec^2 This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually don't have to change it.

Attribute	Axis Type	Data Type	Access	Description
Acceleration Limit Positive	AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units / sec^2 This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually don't have to change it.
Actual Acceleration	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	**Important**: To use this attribute, make sure Auto Tag Update is Enabled for the motion group (default setting). Otherwise, you won't see the right value as the axis runs. Actual Acceleration in Position Units / Sec2 Actual Acceleration is the current instantaneously measured acceleration of an axis, in the configured axis Position Units per second per second. It is calculated as the current increment to the actual velocity per coarse update interval. Actual Acceleration is a signed value — the sign (+ or -) depends on which direction the axis is currently accelerating. Actual Acceleration is a signed floating-point value. Its resolution does not depend on the Averaged Velocity Timebase, but rather on the conversion constant of the axis and the fact that the internal resolution limit on actual velocity is 1 feedback counts per coarse update period per coarse update period.
Actual Position	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	**Important**: To use this attribute, make sure Auto Tag Update is Enabled for the motion group (default setting). Otherwise, you won't see the right value as the axis runs. Actual Position in Position Units Actual Position is the current absolute position of an axis, in the configured Position Units of that axis, as read from the feedback transducer. Note, however, that this value is based on data reported to the controller as part of an ongoing synchronous data transfer process which results in a delay of one coarse update period. Thus, the Actual Position value that is obtained is the actual position of the axis one coarse update period ago.
Actual Velocity	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	**Important**: To use this attribute, make sure Auto Tag Update is Enabled for the motion group (default setting). Otherwise, you won't see the right value as the axis runs. Actual Velocity in Position Units / Sec Actual Velocity is the current instantaneously measured speed of an axis, in the configured axis Position Units per second. It is calculated as the current increment to the actual position per coarse update interval. Actual Velocity is a signed value—the sign (+ or -) depends on which direction the axis is currently moving. Actual Velocity is a signed floating-point value. Its resolution does not depend on the Averaged Velocity Timebase, but rather on the conversion constant of the axis and the fact that the internal resolution limit on actual velocity is 1 feedback counts per coarse update.

Attribute	Axis Type	Data Type	Access	Description
Analog Input1 Analog Input 2	AXIS_SERVO_DRIVE	REAL	GSV SSV	This attribute applies only to an axis associated with a Kinetix 7000 drive. This attribute has an integer range +/-16384 representing the analog value of an analog device connected to the Kinetix 7000 drive analog input. These inputs are useful for web/converting applications with load cell (measuring web force on a roller) or dancer (measuring web force/position directly) that can be directly connected to the drive controlling the web.
Attribute Error Code	AXIS_SERVO AXIS_SERVO_DRIVE	INT	GSV* Tag	CIP Error code returned by erred set attribute list service to the module. When an Axis Configuration Fault occurs, one or more axis parameters associated with a motion module or device has not been successfully updated to match the value of the corresponding parameter of the local controller. The fact that the configuration of the axis no longer matches the configuration of the local controller is a serious fault and results in the shutdown of the faulted axis. The Attribute Error Code is reset to zero by reconfiguration of the motion module. Axis Configuration Fault information is passed from the motion module or device to the controller via a 16-bit CIP status word contained in the Set Attribute List service response received by the controller. A Set Attribute List service to the motion module can be initiated by a software Set Attribute List service to the controller, or by an SSV instruction within the controller's program, referencing a servo attribute. Various routines that process responses to motion services are responsible for updating these attributes. The Set and Get service responses provide a status response with each attribute that was processed. That status value is defined by CIP as follows: UINT16, Values 0-255 (0x00-0xFF) are reserved to mirror common service status codes. Values 256 – 65535 are available for object/class attribute specific errors.
Attribute Error ID	AXIS_SERVO AXIS_SERVO_DRIVE	INT	GSV* Tag	Attribute ID associated with non-zero Attribute Error Code. The Attribute Error ID is used to retain the ID of the servo attribute that returned a non-zero attribute error code resulting in an Axis Configuration Fault. The Attribute Error ID defaults to zero and, after a fault has occurred may be reset to zero by reconfiguration of the motion module. To quickly see the Attribute Error in RSLogix 5000: 1. Select the axis in the Controller Organizer. 2. Look at the bottom of the Controller Organizer for the Attribute Error.

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Aux Feedback Configuration</td><td>AXIS_SERVO_DRIVE</td><td>INT</td><td>GSV</td><td>The controller and drive use this for scaling the feedback device counts. These attributes are derived from the corresponding Motor and Auxiliary Feedback Unit attributes.
Bit
0 = Feedback type
· 0 — rotary (default)
· 1 — linear
1 = (reserved)
2 = Linear feedback unit
· 0 — metric
· 1 — english
3 = Feedback Polarity (Aux Only)
· 0 — not inverted
· 1 — inverted

<table><tr><th colspan="3">If the bits are</th><th>Then Feedback Resolution is scaled to</th></tr><tr><th>2</th><th>1</th><th>0</th><th></th></tr><tr><td>0</td><td></td><td>0</td><td>Feedback Cycles per Feedback Rev</td></tr><tr><td>1</td><td></td><td>0</td><td>Feedback Cycles per Feedback Rev</td></tr><tr><td>0</td><td></td><td>1</td><td>Feedback Cycles per mm</td></tr><tr><td>1</td><td></td><td>1</td><td>Feedback Cycles per inch</td></tr></table>
Feedback Polarity
The Feedback Polarity bit attribute can be used to change the sense of direction of the feedback device. This bit is only valid for auxiliary feedback devices. When performing motor/feedback hookup diagnostics on an auxiliary feedback device using the MRHD and MAHD instructions, the Feedback Polarity bit is configured for the auxiliary feedback device to insure negative feedback into the servo loop. Motor feedback devices must be wired properly for negative feedback since the Feedback Polarity bit is forced to 0, or non-inverted.</td></tr>
<tr><td>Aux Feedback Fault</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>Set for an auxiliary feedback source when one of these happens:
· The differential electrical signals for one or more of the feedback channels (for example, A+ and A-, B+ and B-, or Z+ and Z-) are at the same level (both high or both low). Under normal operation, the differential signals are always at opposite levels. The most common cause of this situation is a broken wire between the feedback transducer and the servo module or drive;
· Loss of feedback “power” or feedback “common” electrical connection between the servo module or drive and the feedback device.
The controller latches this fault. Use a Motion Axis Fault Reset (MAFR) or Motion Axis Shutdown Reset (MASR) instruction to clear the fault.</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Aux Feedback Interpolation Factor	AXIS_SERVO_DRIVE	DINT	GSV	Feedback Counts per Cycle The Feedback Interpolation attributes establish how many Feedback Counts there are in one Feedback Cycle. The Feedback Interpolation Factor depends on both the feedback device and the drive feedback circuitry. Quadrature encoder feedback devices and the associated drive feedback interface typically support 4x interpolation, so the Interpolation Factor for these devices would be set to 4 Feedback Counts per Cycle (Cycles are sometimes called Lines). High Resolution Sin/Cosine feedback device types can have interpolation factors as high as 2048 Counts per Cycle. The product of the Feedback Resolution and the corresponding Feedback Interpolation Factor is the overall resolution of the feedback channel in Feedback Counts per Feedback Unit. In our example, a Quadrature encoder with a 2000 line/rev resolution and 4x interpolation factor would have an overall resolution of 8000 counts/rev.
Aux Feedback Noise Fault	AXIS_SERVO_DRIVE	BOOL	Tag	Set when there is noise on the feedback device's signal lines. · For example, simultaneous transitions of the feedback A and B channels of an A Quad B is referred to generally as feedback noise. · Feedback noise (shown below) is most often caused by loss of quadrature in the feedback device itself or radiated common-mode noise signals being picked up by the feedback device wiring. You can see both of these on an oscilloscope.  · To troubleshoot the loss of channel quadrature, look for: ▪ physical misalignment of the feedback transducer components ▪ excessive capacitance (or other delays) on the encoder signals · Proper grounding and shielding usually cures radiated noise problems. The controller latches this fault. Use a Motion Axis Fault Reset (MAFR) or Motion Axis Shutdown Reset (MASR) instruction to clear the fault.
Aux Feedback Ratio	AXIS_SERVO_DRIVE	FLOAT	GSV	Aux Feedback Units per Motor Feedback Unit The Aux Feedback Ratio attribute represents the quantitative relationship between auxiliary feedback device and the motor. For a rotary auxiliary feedback device, this attribute's value should be the turns ratio between the auxiliary feedback device and the motor shaft. For linear auxiliary feedback devices, this attribute value would typically represent the feed constant between the motor shaft and the linear actuator. The Aux Feedback Ratio attribute is used in calculating range limits and default value calculations during configuration based on the selected motor's specifications. The value is also used by the drive when running the dual feedback servo loop configuration.

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Aux Feedback Resolution</td><td>AXIS_SERVO_DRIVE</td><td>DINT</td><td>GSV</td><td>Cycles per Aux Feedback Unit

The Motor and Aux Feedback Resolution attributes are used to provide the A-B drive with the resolution of the associated feedback device in cycles per feedback unit. These parameters provide the SERCOS drive with critical information needed to compute scaling factors used to convert Drive Counts to Feedback counts.</td></tr>
<tr><td>Aux Feedback Type</td><td>AXIS_SERVO_DRIVE</td><td>INT</td><td>GSV</td><td>The Motor and Aux Feedback Type attributes are used to identify the motor mounted or auxiliary feedback device connected to the drive.
<table>
<tr><th>Feedback Type</th><th>Code</th><th>Rotary Only</th><th>Linear Only</th><th>Rotary or Linear</th></tr>
<tr><td><None></td><td>0x0000</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>SRS</td><td>0x0001</td><td>X</td><td></td><td></td></tr>
<tr><td>SRM</td><td>0x0002</td><td>X</td><td></td><td></td></tr>
<tr><td>SCS</td><td>0x0003</td><td>X</td><td></td><td></td></tr>
<tr><td>SCM</td><td>0x0004</td><td>X</td><td></td><td></td></tr>
<tr><td>SNS</td><td>0x0005</td><td>X</td><td></td><td></td></tr>
<tr><td>MHG</td><td>0x0006</td><td>X</td><td></td><td></td></tr>
<tr><td>Resolver</td><td>0x0007</td><td>X</td><td></td><td></td></tr>
<tr><td>Analog Reference</td><td>0x0008</td><td>X</td><td></td><td></td></tr>
<tr><td>Sin/Cos</td><td>0x0009</td><td></td><td></td><td>X</td></tr>
<tr><td>TTL</td><td>0x000A</td><td></td><td></td><td>X</td></tr>
<tr><td>UVW</td><td>0x000B</td><td></td><td></td><td>X</td></tr>
<tr><td>Unknown Stegmann</td><td>0x000C</td><td></td><td></td><td>X</td></tr>
<tr><td>Endat</td><td>0x000D</td><td></td><td></td><td>X</td></tr>
<tr><td>RCM21S-4</td><td>0x000E</td><td>X</td><td></td><td></td></tr>
<tr><td>RCM21S-6</td><td>0x000F</td><td>X</td><td></td><td></td></tr>
<tr><td>RCM21S-8</td><td>0x0010</td><td>X</td><td></td><td></td></tr>
<tr><td>LINCODER</td><td>0x0011</td><td></td><td>X</td><td></td></tr>
<tr><td>Sin/Cos with Hall</td><td>0x0012</td><td></td><td></td><td>X</td></tr>
<tr><td>TTL with Hall</td><td>0x0013</td><td></td><td></td><td>X</td></tr>
</table></td></tr>
<tr><td>Aux Feedback Units</td><td>AXIS_SERVO_DRIVE</td><td>INT</td><td>GSV</td><td>The Motor Feedback Units attribute establishes the unit of measure that is applied to the Motor Feedback Resolution attribute value. The Aux Feedback Units attribute establishes the unit of measure that is applied to the Aux Feedback Resolution attribute value. Units appearing in the enumerated list cover linear or rotary, english or metric feedback devices.
0 = revs
1 = inches
2 = mm</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Aux Position Feedback	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important**: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. Auxiliary Position Feedback in Position Units Aux Position Feedback is the current value of the position feedback coming from the auxiliary feedback input.
Average Velocity	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	**Important**: To use this attribute, make sure Auto Tag Update is Enabled for the motion group (default setting). Otherwise, you won't see the right value as the axis runs. Average Velocity in Position Units / Sec Average Velocity is the current speed of an axis in the configured Position Units per second of the axis. Unlike the Actual Velocity attribute value, it is calculated by averaging the actual velocity of the axis over the configured Average Velocity Timebase for that axis. Average velocity is a signed value. The sign doesn't necessarily show the direction that the axis is currently moving. It shows the direction the average move is going. The axis may be currently moving in the opposite direction. The resolution of the Average Velocity variable is determined by the current value of the Averaged Velocity Timebase parameter and the configured Conversion Constant (feedback counts per Position Unit) for the axis. · The greater the Average Velocity Timebase value, the better the speed resolution but the slower the response to changes in speed. · The minimum Average Velocity Timebase value is the Coarse Update period of the motion group. The Average Velocity resolution in Position Units per second may be calculated using the equation below. $$\text{Average Velocity Timebase [Seconds]} \times K \left[\frac{\text{Feedback Counts}}{\text{Position Unit}} \right]$$ For example, on an axis with position units of inches and a conversion constant (K) of 20000, an averaged velocity time-base of 0.25 seconds results in an average velocity resolution of: $$\frac{1}{0.25 \times 20000} = 0.0002 \ \frac{\text{Inches}}{\text{Second}} = 0.012 \ \frac{\text{Inches}}{\text{Minute}}$$

Attribute	Axis Type	Data Type	Access	Description
Average Velocity Timebase	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV SSV	Sec The Average Velocity Timebase attribute is used to specify the desired time in seconds to be used for calculating the Average Velocity of the axis. When the Average Velocity Value is requested, the value is computed by taking the total distance traveled by the axis in the amount of time given by the Average Velocity Timebase and dividing this value by the timebase. The Average Velocity Timebase value should be large enough to filter out the small changes in velocity which would otherwise result in a "noisy" velocity value, but small enough to track significant changes in axis velocity. Typically, a value between 0.25 and 0.5 seconds works well for most applications
Axis Address	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL		GSV	Used for debugging.
Axis Configuration State	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	SINT	GSV	State of the axis configuration state machine The Axis Configuration State attribute is used for debugging to indicate where in the axis configuration state-machine this axis presently is. Even consumed and virtual axes will utilize this attribute. If the attribute is: · 128 — the axis is configured and ready for use. · Not 128 — the axis isn't configured.

Attribute	Axis Type	Data Type	Access	Description
Axis Control Bits	AXIS_SERVO AXIS_SERVO_DRIVE	DINT	GSV	Bits 0 = Abort Process Request 1 = Shutdown Request 2 = Zero DAC Request 3 = Abort Home Request 4 = Abort Event Request 5-14 = Reserved 15 = Change Cmd Reference **Abort Process** If this bit is set, any active tuning or test process on the axis is aborted. **Shutdown Request** If this bit is set, the axis is forced into the shutdown state. For an AXIS_SERVO data type, the OK contact opens and the DAC output goes to 0. **Zero DAC Request — Only for AXIS_SERVO Data Type** If this bit is set, the servo module forces the DAC output for the axis to zero volts. This bit only has an affect if the axis is in the Direct Drive State with the drive enabled but no servo action. **Abort Home Request** If this bit is set, any active homing procedures are cancelled. **Abort Event Request** If this bit is set, any active registration or watch event procedures are cancelled. **Change Cmd Reference** If this bit is set, the controller switches to a new position coordinate system for command position. The servo module or drive uses this bit when processing new command position data from the controller to account for the offset implied by the shift in the reference point. The bit is cleared when the axis acknowledges completion of the reference position change by clearing its Change Position Reference bit.

Attribute	Axis Type	Data Type	Access	Description
Axis Data Type	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	SINT	MSG	Associated motion axis tag data type: 0 = Feedback 1 = Consumed 2 = Virtual 3 = Generic 4 = Servo 5 = Servo Drive 6 = Generic Drive The Axis Data Type attribute and is used to determine which data template, memory format, and set of attributes are created and applicable for this axis instance. This attribute can only be set as part of an axis create service. **Feedback** A feedback-only axis associated with feedback-only modules like PLS II and CFE, supporting quadrature encoder, resolver, HiperFace, and so on. **Consumed** A consumed axis which consumes axis motion data produced by a motion axis on another controller. **Virtual** A virtual axis having full motion planner operation but not associated with any physical device. **Generic** An axis with full motion planner functionality but no integrated configuration support; associated with devices such as DriveLogix, 1756-DM. **Servo** An axis with full motion planner functionality and integrated configuration support; associated with modules closing a servo loop and sending an analog command to an external drive; that is, 1756-M02AE, 1756-HYD02, and 1756-M02AS modules. **Servo Drive** An axis with full motion planner functionality and integrated configuration support; associated with digital drive interface modules sending a digital command to the external drive; that is, 1756-M03SE, 1756-M08SE, and 17556-M16SE (SERCOS interface). **Generic Drive** An axis of a SERCOS interface drive that is Extended Pack Profile compliant and on the ring of a 1756-M08SEG module.

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Axis Event</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>DINT</td><td>Tag</td><td>Lets you access all the event status bits in one 32-bit word. This tag is the same as the Axis Event Bits attribute.

<table><tr><th>Event Status</th><th>Bit</th></tr><tr><td>Watch Event Armed Status</td><td>0</td></tr><tr><td>Watch Event Status</td><td>1</td></tr><tr><td>Reg Event 1 Armed Status</td><td>2</td></tr><tr><td>Reg Event 1 Status</td><td>3</td></tr><tr><td>Reg Event 2 Armed Status</td><td>4</td></tr><tr><td>Reg Event 2 Status</td><td>5</td></tr><tr><td>Home Event Armed Status</td><td>6</td></tr><tr><td>Home Event Status</td><td>7</td></tr></table></td></tr>
<tr><td>Axis Event Bits</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>DINT</td><td>GSV</td><td>Lets you access all the event status bits in one 32-bit word. This attribute is the same as the Axis Event tag.

<table><tr><th>Event Status</th><th>Bit</th></tr><tr><td>Watch Event Armed Status</td><td>0</td></tr><tr><td>Watch Event Status</td><td>1</td></tr><tr><td>Reg Event 1 Armed Status</td><td>2</td></tr><tr><td>Reg Event 1 Status</td><td>3</td></tr><tr><td>Reg Event 2 Armed Status</td><td>4</td></tr><tr><td>Reg Event 2 Status</td><td>5</td></tr><tr><td>Home Event Armed Status</td><td>6</td></tr><tr><td>Home Event Status</td><td>7</td></tr></table></td></tr>
<tr><td>Axis Fault</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>DINT</td><td>Tag</td><td>The axis faults for your axis:

<table><tr><th>Type of Fault</th><th>Bit</th></tr><tr><td>Physical Axis Fault</td><td>0</td></tr><tr><td>Module Fault</td><td>1</td></tr><tr><td>Config Fault</td><td>2</td></tr></table>
This attribute is the same as the Axis Fault Bits attribute.</td></tr>
<tr><td>Axis Fault Bits</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>DINT</td><td>GSV*</td><td>The axis faults for your axis:

<table><tr><th>Type of Fault</th><th>Bit</th></tr><tr><td>Physical Axis Fault</td><td>0</td></tr><tr><td>Module Fault</td><td>1</td></tr><tr><td>Config Fault</td><td>2</td></tr></table>
This attribute is the same as the Axis Fault tag.</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Axis Info Select 1 Axis Info Select 2	AXIS_SERVO AXIS_SERVO_DRIVE	DINT	GSV SSV	An axis has a group of attributes that don't get updated by default. · To use one of them, you must choose it for Real Time Axis Information for the axis. Otherwise, its value won't change and you won't see the right value as the axis runs. · You can choose up to 2 of these attributes. To use a GSV instruction to choose an attribute for Real Time Axis Information, set the Axis Info Select 1 or Axis Info Select 2 attribute.

AXIS_SERVO	AXIS_SERVO_DRIVE	Value
None (default)	None (default)	0
Position Command	Position Command	1
Position Feedback	Position Feedback	2
Aux Position Feedback	Aux Position Feedback	3
Position Error	Position Error	4
Position Integrator Error	Position Integrator Error	5
Velocity Command	Velocity Command	6
Velocity Feedback	Velocity Feedback	7
Velocity Error	Velocity Error	8
Velocity Integrator Error	Velocity Integrator Error	9
Acceleration Command	Acceleration Command	10
Acceleration Feedback	Acceleration Feedback	11
Servo Output Level		12
Marker Distance	Marker Distance	13
	Torque Command	14
	Torque Feedback	15
	Positive Dynamic Torque Limit	16
	Negative Dynamic Torque Limit	17
	Motor Capacity	18
	Drive Capacity	19
	Power Capacity	20
	Bus Regulator Capacity	21
	Motor Electrical Angle	22
	Torque Limit Source	23
	DC Bus Voltage	24
Absolute Offset		25

Attribute	Axis Type	Data Type	Access	Description
Axis Instance	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	INT	GSV	Instance Number assigned to Axis The Axis Instance attribute is used to return the instance number of an axis. Major fault records generated for an axis major fault contains only the instance of the offending axis. This attribute would then typically be used by a user to determine if this was the offending axis; that is, if the instance number matches.
Axis Response Bits	AXIS_SERVO AXIS_SERVO_DRIVE	DINT	GSV	Bits 0 = Abort Process Acknowledge 1 = Shutdown Acknowledge 2 = Zero DAC Acknowledge 3 = Abort Home Acknowledge 4 = Abort Event Acknowledge 5-14 = Reserved 15 = Change Pos Reference **Abort Process Acknowledge** If this bit is set, the tuning or test process has been aborted. **Shutdown Acknowledge** If this bit is set, the axis has been forced into the shutdown state. **Zero DAC Acknowledge — Only for AXIS_SERVO Data Type** If this bit is set, the DAC output for the axis has been set to zero volts. **Abort Home Acknowledge** If this bit is set, the active home procedure has been aborted. **Abort Event Acknowledge** If this bit is set, the active registration or watch position event procedure has been aborted. **Change Pos Reference** If this bit is set, the Servo loop has switched to a new position coordinate system. The controller uses this bit when processing new position data from the servo module or drive to account for the offset implied by the shift in the reference point. The bit is cleared when the conroller acknowledges completion of the reference position change by clearing its Change Cmd Reference bit.

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Axis State</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>SINT</td><td>GSV</td><td>Operating state of the axis.
0 = Axis Ready
1 = Direct Drive Control
2 = Servo Control
3 = Axis Faulted
4 = Axis Shutdown
5 = Axis Inhibited
6 = Axis Ungrouped
7 = No Module
8 = Configuring</td></tr>
<tr><td>Axis Status</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>DINT</td><td>Tag</td><td>Lets you access all the axis status bits in one 32-bit word. This tag is the same as the Axis Status Bits attribute.
<table><tr><th>Axis Status</th><th>Bit</th></tr><tr><td>Servo Action Status</td><td>0</td></tr><tr><td>Drive Enable Status</td><td>1</td></tr><tr><td>Shutdown Status</td><td>2</td></tr><tr><td>Config Update In Process</td><td>3</td></tr><tr><td>Inhibit Status</td><td>4</td></tr></table></td></tr>
<tr><td>Axis Status Bits</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>DINT</td><td>GSV*</td><td>Lets you access all the axis status bits in one 32-bit word. This attribute is the same as the Axis Status tag.
<table><tr><th>Axis Status</th><th>Bit</th></tr><tr><td>Servo Action Status</td><td>0</td></tr><tr><td>Drive Enable Status</td><td>1</td></tr><tr><td>Shutdown Status</td><td>2</td></tr><tr><td>Config Update In Process</td><td>3</td></tr><tr><td>Inhibit Status</td><td>4</td></tr></table></td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Axis Type</td><td>AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>INT</td><td>GSV
SSV</td><td>The Axis Type attribute is used to establish the intended use of the axis.

<table><tr><th>If</th><th>Then set the attribute to</th></tr><tr><td>The axis is unused in the application, which is a common occurrence when there are an odd number of axes in the system</td><td>0</td></tr><tr><td>You only want the position information from the feedback interface</td><td>1</td></tr><tr><td>The axis is intended for full servo operation</td><td>2</td></tr></table>
Axis Type is not only used to qualify many operations associated with the axis servo loop, it also controls the behavior of the servo module's Axis Status LEDs. An Axis Type of "1" (Feedback Only) results in the DRIVE LED being blanked, while a value of "0" (Unused) blanks both the FDBK and DRIVE LEDs.

RSLogix 5000 software also uses the current configured value for Axis Type to control the look of many of the dialogs associated with configurating an axis.</td></tr>
<tr><td>Backlash Reversal Offset</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
SSV</td><td>Backlash Reversal Offset provides the user the capability to compensate for positional inaccuracy introduced by mechanical backlash. For example, power-train type applications require a high level of accuracy and repeatability during machining operations. Axis motion is often generated by a number of mechanical components such as a motor, a gearbox, and a ball-screw, which can introduce inaccuracies and which are subject to wear over their lifetime. Hence, when an axis is commanded to reverse direction, mechanical play in the machine (through the gearing, ball-screw, and so on.) may result in a small amount of motor motion without axis motion. As a result, the feedback device may indicate movement even though the axis has not physically moved.

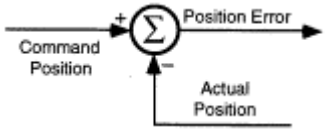
Compensation for mechanical backlash can be achieved by adding a directional offset, specified by the Backlash Reversal Offset attribute, to the motion planner's command position as it is applied to the associated servo loop. Whenever the commanded velocity changes sign (a reversal), the Logix controller adds, or subtracts, the Backlash Distance value from the current commanded position. This causes the servo to immediately move the motor to the other side of the backlash window and engage the load. It is important to note that the application of this directional offset is completely transparent to the user; the offset does not have any affect on the value of the Command Position attribute.

If a value of zero is applied to the Backlash Reversal Offset, the feature is effectively disabled. Once enabled by a non-zero value, and the load is engaged by a reversal of the commanded motion, changing the Backlash Reversal Offset can cause the axis to shift as the offset correction is applied to the command position..</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Backlash Stabilization Window	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	The Backlash Stabilization Window attribute is used to control the Backlash Stabilization feature in the servo control loop. What follows is a description of this feature and the general backlash instability phenomenon. Mechanical backlash is a common problem in applications that utilize mechanical gearboxes. The problem stems from the fact that until the input gear is turned to the point where its proximal tooth contacts an adjacent tooth of the output gear, the reflected inertia of the output is not felt at the motor. In other words, when the gear teeth are not engaged, the system inertia is reduced to the motor inertia. If the servo loop is tuned for peak performance with the load applied, the axis is at best under-damped and at worst unstable in the condition where the gear teeth are not engaged. In the worst case scenario, the motor axis and the input gear oscillates wildly between the limits imposed by the output gear teeth. The net effect is a loud buzzing sound when the axis is at rest. If this situation persists the gearbox wears out prematurely. To prevent this condition, the conventional approach is to de-tune the servo so that the axis is stable without the gearbox load applied. Unfortunately, system performance suffers." Due to its non-linear, discontinuous nature, adaptive tuning algorithms generally fall short of addressing the backlash problem. However, a very effective backlash compensation algorithm can be demonstrated using the Torque Scaling gain. The key to this algorithm is the tapered Torque Scaling profile as a function of the position error of the servo loop. The reason for the tapered profile, as opposed to a step profile, is that when the position error exceeds the backlash distance a step profile would create a very large discontinuity in the torque output. This repulsing torque tends to slam the axis back against the opposite gear tooth and perpetuate the buzzing effect. The tapered Torque Scaling profile is only run when the acceleration command to the servo loop is zero, that is, when we are not commanding any acceleration or deceleration that would engage the teeth of the gearbox. Properly configured with a suitable value for the Backlash Stabilization Window, this algorithm entirely eliminates the gearbox buzz without sacrificing any servo performance. The Backlash Stabilization parameter determines the width of the window over which backlash stabilization is applied. In general, this value should be set to the measured backlash distance. A Backlash Stabilization Window value of zero effectively disables the feature. (Patent Pending)

Attribute	Axis Type	Data Type	Access	Description
Brake Engage Delay Time	AXIS_SERVO_DRIVE	REAL	GSV SSV	Sec The Brake Engage Delay attribute controls the amount of time that the drive continues to apply torque to the motor after the motor brake output is changed to engage the brake. This gives time for the motor brake to engage. This is the sequence of events associated with engaging the motor brake. · Disable axis is initiated (via MSF or drive disable fault action) · Drive stops tracking command reference. (Servo Action Status bit clears.) · Decel to zero speed using configured Stopping Torque. · Zero speed or Stopping Time Limit is reached. · Turn motor brake output off to engage the motor brake. · Wait Brake Engage Delay Time. · Disable the drive power structure. (Drive Enable Status bit clears.) If the axis is shutdown through either a fault action or motion instruction the drive power structure is disabled immediately and the motor brake is engaged immediately. · Drive stops tracking command reference. (Servo Action Status bit clears.) · Disable drive power structure, (Drive Enable Status bit clears.) · Turn off brake output to engage brake.
Brake Release Delay Time	AXIS_SERVO_DRIVE	REAL	GSV SSV	Sec The Brake Release Delay attribute controls the amount of time that the drive holds off tracking command reference changes after the brake output is changed to release the brake. This gives time for the brake to release. This is the sequence of events associated with engaging the brake. · Enable axis is initiated (via MSO or MAH) · Drive power structure enabled. (Drive Enable Status bit sets.) · Turn motor brake output on to release the brake.** · Wait Brake Release Delay Time. · Track command reference. (Servo_Action_Status bit sets) **The drive does not release the brake unless there is holding torque.
Bus Ready Status	AXIS_SERVO_DRIVE	BOOL	Tag	If the bit is: · ON — The voltage of the drive's dc bus is high enough for operation. · OFF — The voltage of the drive's dc bus is too low.

Attribute	Axis Type	Data Type	Access	Description
Bus Regulator Capacity	AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important**: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. The present utilization of the axis bus regulator as a percent of rated capacity.
Bus Regulator ID	AXIS_SERVO_DRIVE	INT	GSV	The Bus Regulator ID attribute contains the enumeration of the specific A-B Bus Regulator or System Shunt catalog numbers associated with the axis. If the Bus Regulator ID does not match that of the actual bus regulator or shunt hardware, an error is generated during the drive configuration process.
C2C Connection Instance	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	SINT	GSV	Producer/Consumed axis's associated C2C connection instance in reference to the C2C map instance When Axis Data Type is specified to be 'Consumed' then this axis is associated to the consumed data by specifying both the C2C Map Instance and the C2C Connection Instance. This attribute is the connection instance under the C2C map instance, which provides the axis data being sent to it from another axis via a C2C connection. For all other Axis Data Types if this axis is to be produced then this attribute is set to the connection instance under the local controller's map instance (1) that is used to send the remote axis data via the C2C connection.
C2C Map Instance	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	SINT	GSV	Producer/Consumed axis's associated C2C map instance When the Axis Data Type attribute is specified to be 'Consumed' then this axis is associated to the consumed data by specifying both the C2C Map Instance and the C2C Connection Instance. For all other Axis Data Types if this axis is to be produced then this attribute is set to 1 (one) to indicate that the connection is off of the local controller's map instance.
Command Acceleration	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	**Important**: To use this attribute, make sure Auto Tag Update is Enabled for the motion group (default setting). Otherwise, you won't see the right value as the axis runs. Command Acceleration in Position Units / Sec2 Command Acceleration is the commanded speed of an axis, in the configured axis Position Units per second per second, as generated by any previous motion instructions. It is calculated as the current increment to the command velocity per coarse update interval. Command Acceleration is a signed value—the sign (+ or -) depends on which direction the axis is being commanded to move. Command Acceleration is a signed floating-point value. Its resolution does not depend on the Averaged Velocity Timebase, but rather on the conversion constant of the axis and the fact that the internal resolution limit on command velocity is 0.00001 feedback counts per coarse update period per coarse update period.

Attribute	Axis Type	Data Type	Access	Description
Command Position	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	**Important**: To use this attribute, make sure Auto Tag Update is Enabled for the motion group (default setting). Otherwise, you won't see the right value as the axis runs. Command Position in Position Units Command Position is the desired or commanded position of a physical axis, in the configured Position Units of that axis, as generated by the controller in response to any previous motion Position Control instruction. Command Position data is transferred by the controller to a physical axis as part of an ongoing synchronous data transfer process which results in a delay of one coarse update period. Thus, the Command Position value that is obtained is the command position that is acted upon by the physical servo axis one coarse update period from now. The figure below shows the relationship between Actual Position, Command Position, and Position Error for an axis with an active servo loop. Actual Position is the current position of the axis as measured by the feedback device (for example, encoder). Position error is the difference between the Command and Actual Positions of the servo loop, and is used to drive the motor to make the actual position equal to the command position.  Command position is useful when performing motion calculations and incremental moves based on the current position of the axis while the axis is moving. Using command position rather than actual position avoids the introduction of cumulative errors due to the position error of the axis at the time the calculation is performed.
Command Velocity	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	**Important**: To use this attribute, make sure Auto Tag Update is Enabled for the motion group (default setting). Otherwise, you won't see the right value as the axis runs. Command Velocity in Position Units / Sec Command Velocity is the commanded speed of an axis, in the configured axis Position Units per second, as generated by any previous motion instructions. It is calculated as the current increment to the command position per coarse update interval. Command Velocity is a signed value—the sign (+ or -) depends on which direction the axis is being commanded to move. Command Velocity is a signed floating-point value. Its resolution does not depend on the Averaged Velocity Timebase, but rather on the conversion constant of the axis and the fact that the internal resolution limit on command velocity is 0.00001 feedback counts per coarse update.
Common Bus Fault	AXIS_SERVO_DRIVE	BOOL	Tag	The drive shuts down if you give it 3-phase power while it's configured for Common Bus Follower mode. If that happens, this bit turns on.

Attribute	Axis Type	Data Type	Access	Description
Commutation Fault	AXIS_SERVO_DRIVE	DINT	BOOL	Set when the commutation feedback source associated with the drive axis has a problem that prevents the drive from receiving accurate or reliable motor shaft information to perform commutation.
Config Fault	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set when an update operation targeting an axis configuration attribute of an associated motion module has failed. Specific information concerning the Configuration Fault may be found in the Attribute Error Code and Attribute Error ID attributes associated with the motion module. Do you want this fault to give the controller a major fault? · YES — Set the General Fault Type of the motion group = Major Fault. · NO — You must write code to handle these faults.
Config Update In Process	AXIS_CONSUMED AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	When you use an SSV instruction to change an attribute, the controller sends the change to the motion module. If you want to wait until the change is done, monitor the ConfigUpdateInProcess bit of the axis. If the bit is: · ON — The controller is changing the attribute. · OFF — The change is done.
Continuous Torque Limit	AXIS_SERVO_DRIVE	REAL	GSV SSV	%Rated The Torque limit attribute provides a method for controlling the continuous torque limit imposed by the drive's thermal model of the motor. Increasing the Continuous Torque Limit increases the amount of continuous motor torque allowed before the drive either folds back the motor current or the drive declares a motor thermal fault. Motors equipped with special cooling options can be configured with a Continuous Torque Limit of greater than 100% rated to attain higher continuous torque output from the motor. Motors operating in high ambient temperature conditions can be configured with a Continuous Torque Limit of less than 100% rated torque to protect the motor from overheating. The Continuous Torque Limit specifies the maximum percentage of the motor's rated current that the drive can command on a continuous or RMS basis. For example, a Continuous Torque Limit of 150% limits the continuous current delivered to the motor to 1.5 times the continuous current rating of the motor.
Control Sync Fault	AXIS_CONSUMED AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	If this bit is set, the controller lost communication with the motion module and missed several position updates in a row. · The controller can miss up to 4 position updates. After that, the Control Sync Fault bit is set. The motion module may fault later or may already be faulted. · For a consumed axis, this bit means that communication is lost with the producing controller. This bit clears when communication is reestablished.

Attribute	Axis Type	Data Type	Access	Description
Controlled By Transform Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	If the bit is: · ON — A transform is moving the axis. · OFF — A transform isn't moving the axis.
Conversion Constant	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV SSV	Counts / Position Unit Range = 0.1 - $1e^{12}$ Default = 8000.0 To allow axis position to be displayed and motion to be programmed in the position units specified by the Position Unit string attribute, a Conversion Constant must be established for each axis. The Conversion Constant, sometimes known as the K constant, allows the Axis Object to convert the axis position units into feedback counts and vice versa. Specifically, K is the number of feedback counts per Position Unit. Note that the 1756M02AE encoder based servo module uses 4X encoder feedback decoding (both edges of channel A and B are counted). The count direction is determined from both the direction of the edge and the state of the opposite channel. Channel A leads channel B for increasing count. This is the most commonly used decode mode with incremental encoders, since it provides the highest resolution. For example, suppose this servo axis utilizes a 1000 line encoder in a motor coupled directly to a 5 pitch lead screw (5 turns per inch). With a user defined Position Unit of Inches, the conversion constant is calculated as shown below: K = 1000 Lines/Rev * 4 Counts/Line * 5 Revs/Inch = 20,000 Counts/Inch. **Attention**: If 'Conversion Constant' is changed it invalidates all of the settable attributes with "Position Unit" conversions in "Description" column. To be valid the 'Conversion Constant' must be set to the desired value prior to setting (including defaulting) any of the affected attributes.

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Coordinated Motion Status</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>DINT</td><td>GSV
Tag</td><td>Set if any coordinated motion profile is currently active upon the axis. It is cleared as soon as Coordinated Motion is complete or stopped.
<table><tr><th>Motion Status</th><th>Bit</th></tr><tr><td>Accel Status</td><td>0</td></tr><tr><td>Decel Status</td><td>1</td></tr><tr><td>Actual Pos Tolerance Status</td><td>2</td></tr><tr><td>Command Pos Tolerance Status</td><td>3</td></tr><tr><td>Stopping Status</td><td>4</td></tr><tr><td>Reserved</td><td>5</td></tr><tr><td>Move Status</td><td>6</td></tr><tr><td>Transition Status</td><td>7</td></tr><tr><td>Move Pending Status</td><td>8</td></tr><tr><td>Move Pending Queue Full Status</td><td>9</td></tr><tr><td>Reserved</td><td>10</td></tr><tr><td>Reserved</td><td>11</td></tr><tr><td>Reserved</td><td>12</td></tr><tr><td>Coordinate System in a Source CS</td><td>13</td></tr><tr><td>Coordinate System in a Target CS</td><td>14</td></tr></table></td></tr>
<tr><td>Damping Factor</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
SSV</td><td>The Damping Factor attribute value is used in calculating the maximum Position Servo Bandwidth (see below) during execution of the MRAT (Motion Run Axis Tune) instruction. In general the Damping Factor attribute controls the dynamic response of the servo axis. When gains are tuned using a small damping factor (like 0.7), a step response test performed on the axis would demonstrate under-damped behavior with velocity overshoot. A gain set generated using a larger damping factor, like 1.0, would produce a system step response that has no overshoot but has a significantly lower servo bandwidth. The default value for the Damping Factor of 0.8 should work fine for most applications.</td></tr>
<tr><td>DC Bus Voltage</td><td>AXIS_SERVO_DRIVE</td><td>DINT</td><td>GSV
Tag</td><td>Important: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1.

Volts

This parameter is the present voltage on the DC Bus of the drive.</td></tr>
<tr><td>Decel Status</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>BOOL</td><td>Tag</td><td>Set if the axis is currently being commanded to decelerate.
Use the Accel Status bit and the Decel Status bit to see if the axis is accelerating or decelerating. If both bits are off, then the axis is moving at a steady speed or is at rest.</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Direct Drive Ramp Rate	AXIS_SERVO	REAL	GSV SSV	Volts/Second The Direct Drive Ramp Rate attribute contains a slew rate for changing the output voltage when the Direct Drive On (MDO) instruction is executed. A Direct Drive Ramp Rate of 0, disables the output ramp rate limiter, allowing the Direct Drive On voltage to be applied directly.
Directional Scaling Ratio	AXIS_SERVO	REAL	GSV SSV	In some cases, the speed or velocity scaling of the external drive actuator may be directionally dependent. This non-linearity can be substantial in hydraulic applications. To compensate for this behavior, the Directional Scaling Ratio attribute can be applied to the Velocity Scaling based on the sign of the Servo Output. Specifically, the Velocity Scaling value is scaled by the Directional Scaling Ratio when the sign of the Servo Output is positive. Thus, the Directional Scaling Ratio is the ratio of the Velocity Scaling in the positive direction (positive servo output) to the Velocity Scaling in the negative direction (negative servo output). The value for the Directional Scaling ratio can be empirically determined by running the auto-tune procedure in the positive direction and then in the negative direction and calculating the ratio of the resulting Velocity/Torque Scaling values.
Drive Axis ID	AXIS_SERVO_DRIVE	INT	GSV	Product Code of Drive Amplifier The Drive ID attribute contains the ASA Product Code of the drive amplifier associated with the axis. If the Product Code does not match that of the actual drive amplifier, an error is generated during the configuration process.
Drive Capacity	AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important**: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. The present utilization of drive capacity as a percent of rated capacity.
Drive Control Voltage Fault	AXIS_SERVO_DRIVE	BOOL	Tag	Set when the power supply voltages associated with the drive circuitry fall outside of acceptable limits.
Drive Cooling Fault	AXIS_SERVO_DRIVE	BOOL	Tag	Set when the ambient temperature surrounding the drive's control circuitry temperature exceeds the drive ambient shut-down temperature.

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Drive Enable Input Fault</td><td>AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>This fault would be declared if either one of two possible conditions occur: 1) If an attempt is made to enable the axis (typically via MSO or MAH instruction) while the Drive Enable Input is inactive. 2) If the Drive Enable Input transitions from active to inactive while the axis is enabled.

This fault can only occur when the Drive Enable Input Fault Handling bit is set in the Fault Configuration Bits attribute.

If the Drive Enable Input Fault Action is set for Stop Command and the axis is stopped as a result of a Drive Enable Input Fault, the faulted axis cannot be moved until the fault is cleared. Any attempt to move the axis in the faulted state using a motion instruction results in an instruction error.

If the Drive Enable Fault Action setting is Status Only or Stop Command and an attempt is made to enable the axis (typically via MSO or MAH instruction) while the Drive Enable Input is active, the axis enables in the faulted state indicating a Drive Enable Input Fault. When the Drive Enable Fault Action setting is Stop Command, instructions that both enable the axis and initiate motion (MAH, MRAT, MAHD) abort the motion process leaving the instruction with both the IP and PC bits clear.

This fault condition is latched and requires execution of an explicit MAFR (Motion Axis Fault Reset) or MASR (Motion Axis Shutdown Reset) instruction to clear. Any attempt to clear the fault while the drive enable input is still inactive and the drive is enabled is unsuccessful. However, the drive enable input fault may be cleared with the drive enable input inactive if the drive is disabled.

If the Drive Enable Input Checking bit is clear, then the state of the Drive Enable Input is irrelevant so no fault would be declared in any of the above conditions.</td></tr>
<tr><td>Drive Enable Input Fault Action</td><td>AXIS_SERVO_DRIVE</td><td>SINT</td><td>GSV
SSV</td><td><table><tr><th>Fault Action</th><th>Value</th></tr><tr><td>Shutdown</td><td>0</td></tr><tr><td>Disable Drive</td><td>1</td></tr><tr><td>Stop Motion</td><td>2</td></tr><tr><td>Status Only</td><td>3</td></tr></table></td></tr>
<tr><td>Drive Enable Status</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>BOOL</td><td>Tag</td><td>AXIS_SERVO
If this bit is:
· ON — The Drive Enable output of the axis is on.
· OFF — Drive Enable output of the axis is off.

AXIS_SERVO_DRIVE
If this bit is:
· ON — The drive's power structure is active.
· OFF — The drive's power structure is not active.</td></tr>
<tr><td>Drive Fault</td><td>AXIS_SERVO</td><td>BOOL</td><td>Tag</td><td>If this bit is set, then the external servo drive has detected a fault and has communicated the existence of this fault to the servo module via the Drive Fault input. This fault condition is latched and requires execution of an explicit MAFR (Motion Axis Fault Reset) or MASR (Motion Axis Shutdown Reset) instruction to clear.</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Drive Fault	AXIS_SERVO_DRIVE	DINT	Tag	Lets you access all the drive fault bits in one 32-bit word. This tag is the same as the Drive Fault Bits attribute.

Tag	Bit
Pos Soft Overtravel Fault	0
Neg Soft Overtravel Fault	1
Pos Hard Overtravel Fault	2
Neg Hard Overtravel Fault	3
Mot Feedback Fault	4
Mot Feedback Noise Fault	5
Aux Feedback Fault	6
Aux Feedback Noise Fault	7
Reserved	8
Drive Enable Input Fault	9
Common Bus Fault	10
Precharge Overload Fault	11
Guard Fault Exists	12
Ground Short Fault	13
Drive Hard Fault	14
Overspeed Fault	15
Overload Fault	16
Drive Overtemp Fault	17
Motor Overtemp Fault	18
Drive Cooling Fault	19
Drive Control Voltage Fault	20
Feedback Fault	21
Commutation Fault	22
Drive Overcurrent Fault	23
Drive Overvoltage Fault	24
Drive Undervoltage Fault	25
Power Phase Loss Fault	26
Position Error Fault	27
SERCOS Fault	28
Overtravel Fault	29
Reserved	30
Manufacturer Specific Fault	31

Do you want any of these faults to give the controller a major fault?

- YES - Set the General Fault Type of the motion group = Major Fault.
- NO - You must write code to handle these faults.

For more information about error codes displayed on drives and/or multi-axis motion control systems, see page 385.

Attribute	Axis Type	Data Type	Access	Description
Drive Fault Action	AXIS_SERVO	SINT	GSV SSV	**Fault Action** / **Value** Shutdown / 0 Disable Drive / 1 Stop Motion / 2 Status Only / 3

Attribute	Axis Type	Data Type	Access	Description
Drive Fault Bits	AXIS_SERVO_DRIVE	DINT	GSV	Lets you access all the drive fault bits in one 32-bit word. This attribute is the same as the Drive Fault tag.

Tag	Bit
Pos Soft Overtravel Fault	0
Neg Soft Overtravel Fault	1
Pos Hard Overtravel Fault	2
Neg Hard Overtravel Fault	3
Mot Feedback Fault	4
Mot Feedback Noise Fault	5
Aux Feedback Fault	6
Aux Feedback Noise Fault	7
Reserved	8
Drive Enable Input Fault	9
Common Bus Fault	10
Precharge Overload Fault	11
GuardFaultExists	12
Ground Short Fault	13
Drive Hard Fault	14
Overspeed Fault	15
Overload Fault	16
Drive Overtemp Fault	17
Motor Overtemp Fault	18
Drive Cooling Fault	19
Drive Control Voltage Fault	20
Feedback Fault	21
Commutation Fault	22
Drive Overcurrent Fault	23
Drive Overvoltage Fault	24
Drive Undervoltage Fault	25
Power Phase Loss Fault	26
Position Error Fault	27
SERCOS Fault	28
Overtravel Fault	29
Reserved	30
Manufacturer Specific Fault	31

Do you want any of these faults to give the controller a major fault?

- YES — Set the General Fault Type of the motion group = Major Fault.
- NO — You must write code to handle these faults.

Attribute	Axis Type	Data Type	Access	Description
Drive Fault Input Status	AXIS_SERVO	BOOL	Tag	Digital output from the drive that shows if there is a fault. If this bit is: · ON — The drive is has a fault. · OFF — The drive doesn't have a fault.
Drive Hard Fault	AXIS_SERVO_DRIVE	BOOL	Tag	Set when the drive detects a serious hardware fault.
Drive Model Time Constant	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	Sec The value for the Drive Model Time Constant represents the lumped model time constant for the drive's current loop used by the MRAT instruction to calculate the Maximum Velocity and Position Servo Bandwidth values. The Drive Model Time Constant is the sum of the drive's current loop time constant, the feedback sample period, and the time constant associated with the velocity feedback filter. This value is set to a default value when you configure the axis.

For this Axis type	Details
AXIS_SERVO	This value is only used by MRAT when the axis is configured for an External Torque Servo Drive..
AXIS_SERVO_DRIVE	Since the bandwidth of the velocity feedback filter is determined by the resolution of the feedback device, the value for the Drive Model Time Constant is smaller when high resolution feedback devices are selected.

Drive Overcurrent Fault	AXIS_SERVO_DRIVE	BOOL	Tag	Set when drive output current exceeds the predefined operating limits for the drive.
Drive Overtemp Fault	AXIS_SERVO_DRIVE	BOOL	Tag	Set when the drive's temperature exceeds the drive shutdown temperature.
Drive Overvoltage Fault	AXIS_SERVO_DRIVE	BOOL	Tag	Set when drive DC bus voltage exceeds the predefined operating limits for the bus.

Attribute	Axis Type	Data Type	Access	Description
Drive Polarity	AXIS_SERVO_DRIVE	DINT	GSV SSV	0 = Custom Polarity 1 = Positive Polarity 2 = Negative Polarity **Custom Polarity** Custom Polarity is used to enable custom polarity configurations using the various polarity parameters defined by the SERCOS Interface standard. **Positive/Negative Polarity** Positive and Negative Polarity bit attribute determines the overall polarity of the servo loop of the drive. All the advanced polarity parameters are automatically set based on whether the Drive Polarity is configured as Positive or Negative. Proper wiring guarantees that the servo loop is closed with negative feedback. However there is no such guarantee that the servo drive has the same sense of forward direction as the user for a given application. Negative Polarity inverts the polarity of both the command position and actual position data of the servo drive. Thus, selecting either Positive or Negative Drive Polarity makes it possible to configure the positive direction sense of the drive to agree with that of the user. This attribute is configured automatically using the MRHD and MAHD motion instructions. Refer to the Logix Motion Instruction Specification for more information on these hookup diagnostic instructions.

Attribute	Axis Type	Data Type	Access	Description
Drive Resolution	AXIS_SERVO_DRIVE	DINT	GSV	Drive Counts / Drive Unit The Drive Resolution attribute determines how many Drive Counts there are in a Drive Unit. Drive Units may be configured as Revs, Inches, or Millimeters depending on the specific drive application. Furthermore, the configured Drive Unit may apply to either a motor or auxiliary feedback device. All position, velocity, and acceleration data to the drive is scaled from the user's Position Units to Drive Units based on the Drive Resolution and Conversion Constant. The ratio of the Conversion Constant to Drive Resolution determines the number of Position Units in a Drive Unit. Conversion Constant / Drive Resolution = Drive Units (rev, inch, or mm) / Position Unit Conversely, all position, velocity, and acceleration data from the drive is scaled from the user's Position Units to Drive Units based on the Drive Resolution and Conversion Constant. The ratio of Drive Resolution and the Conversion Constant determines the number of Position Units in a Drive Unit. Drive Resolution / Conversion Constant = Position Units / Drive Unit (rev, inch, or mm) In general, the Drive Resolution value may be left at its default value of 200000 Drive Counts per Drive Unit, independent of the resolution of the feedback device(s) used by the drive. This is because the drive has its own set of scale factors that it uses to relate feedback counts to drive counts. **Drive Travel Range Limit** Because the drive's position parameters are ultimately limited to signed 32-bit representation per the SERCOS standard, the Drive Resolution parameter impacts the drive's travel range. The equation for determining the maximum travel range based on Drive Resolution is as follows: Drive Travel Range Limit = +/- 2,147,483,647 / Drive Resolution. Based on a default value of 200,000 Drive Counts per Drive Unit, the drive's range limit is 10,737 Drive Units. While it is relatively rare for this travel range limitation to present a problem, it is a simple matter to lower the Drive Resolution to increase the travel range. The downside of doing so is that the position data is then passed with lower resolution that could affect the smoothness of motion. **Fractional Unwind** In some cases, however, the user may also want to specifically configure Drive Resolution value to handle fractional unwind applications or multi-turn absolute applications requiring cyclic compensation. In these cases where the Unwind value for a rotary application does not work out to be an integer value, the Rotational Position Scaling attribute may be modified to a value that is integer divisible by the Unwind value. The following examples demonstrate how the Drive Resolution value may be used together with the Conversion Constant to handle various applications. ***Continued on next page***

Attribute	Axis Type	Data Type	Access	Description
Drive Resolution (cont.)				**Rotary Gear-Head WITHOUT Aux Feedback Device** Based on a rotary motor selection, Drive Resolution would be expressed as Drive Counts per Motor Rev and be applied to the Rotational Position Resolution IDN. The user would set the Conversion Constant to Drive Counts per user-defined Position Unit. If it is a 3:1 gearbox, and the user's Position Unit is, say, Revs of the gear output shaft, the Conversion Constant is 200,000/3, which is irrational! But, in this case, you could simply set the Drive Resolution to 300,000 Drive Counts/Motor Rev and the Conversion Constant could then be set to 100,000 Drive Counts/Output Shaft Rev. This system would work with this configuration without any loss of mechanical precision, that is, a move of 1 output shaft revolution would move the output shaft exactly 1 revolution. **Linear Ball-Screw WITHOUT Aux Feedback Device** Based on a rotary motor selection, Drive Resolution would be expressed as Drive Counts per Motor Rev and be applied to the Rotational Position Resolution IDN. The user would set the Conversion Constant to Drive Counts per user-defined Position Unit. If it is a 5mm pitch ball-screw, and the user's Position Unit is, say, mm, the user simply sets the Conversion Constant to 200,000/5 or 40,000 Drive Counts per mm based on the default Drive Resolution value of 200,000 Drive Counts/Motor Rev. If the pitch is irrational, the method for addressing this is the same as described in Rotary Gear-Head WITHOUT Aux Feedback Device. **Rotary Gear-Head WITH Aux Feedback Device** Based on a rotary motor feedback selection, Drive Resolution would be expressed as Drive Counts per Aux Rev and be applied to the Rotational Position Resolution IDN. Now that position is based on the auxiliary feedback device according to the Servo Loop Configuration, the Data Reference bit of the various Scaling Types should be Load Referenced rather than Motor Referenced. The motor feedback would be rotary and resolution expressed in cycles per motor rev. The aux feedback device is also rotary and its resolution expressed in cycles per aux rev. The Aux Feedback Ratio would be set to the number of aux feedback revs per motor rev and internally applied to IDNs 121 and 122 for the purpose of relating position servo loop counts to velocity servo loop counts in a dual servo loop configuration. The Aux Feedback Ratio attribute is also used in range limit and default value calculations during configuration based on the selected motor's specifications. If the application uses a 3:1 gearbox, and the user's Position Unit is, say, Revs of the gearbox output shaft, the Conversion Constant is still rational, since our scaling is Load Referenced! The user simply sets the Conversion Constant to 200,000 Drive Counts/Output Shaft Rev based on the default Drive Resolution value of 200,000 Drive Counts/Aux Rev. The system would work in this configuration without any loss of mechanical precision, that is, a move of 1 output shaft revolution would move the output shaft exactly 1 revolution. ***Continued on next page***

Attribute	Axis Type	Data Type	Access	Description
Drive Resolution (cont.)				**Linear Ball-Screw/Ball-Screw Combination WITH Aux Feedback Device** Based on a linear aux feedback selection, Drive Resolution would be expressed as Drive Counts per Linear Unit, say Millimeters (Metric bit selection), and be applied to the Linear Position Data Scaling IDNs. Now that position is based on the auxiliary feedback device according to the Servo Loop Configuration, the Data Reference bit of the various Scaling Types should again be Load Referenced rather than Motor Referenced. The motor feedback would be rotary and resolution expressed in cycles per motor rev. The aux feedback device is now linear and its resolution expressed in cycles per, say, mm. The Aux Feedback Ratio would be set to the number of aux feedback units (mm) per motor rev and internally applied to IDN 123 to relate position servo loop counts to velocity servo loop counts in a dual servo loop configuration. The Aux Feedback Ratio attribute is also used in range limit and default value calculations during configuration based on the selected motor's specifications. If the application uses a 3:1 gearbox and a 5 mm pitch ball-screw, and the user's Position Unit is, say, cm, the Conversion Constant is again rational, since we are Load Referenced! The user sets the Conversion Constant to 20,000 Drive Counts/cm based on the default Drive Resolution value of 200000 Drive Counts/mm. This system would work in this configuration without any loss of mechanical precision, that is, a move of 10 cm would move the actuator exactly 10 cm.

Attribute	Axis Type	Data Type	Access	Description
Drive Scaling Bits	AXIS_SERVO_DRIVE	DINT	GSV	The Drive Scaling Bits attribute configuration is derived directly from the Drive Units attribute. Bits 0 = Scaling type 0 – standard 1 – custom 1 = Scaling unit 0 – rotary 1 – linear 2 = Linear scaling unit 0 – metric 1 – english 3 = Data Reference 0 – motor 1 – load **Scaling Type** The Scaling Type bit attribute is used to enable custom scaling using the position, velocity, acceleration, and torque scaling parameters defined by the SERCOS Interface standard. When the bit is clear (default), these scaling parameters are all set based on the preferred Rockwell Automation SERCOS drive scaling factors. Currently there is no Logix support for custom scaling. **Scaling Unit** The Scaling Unit attribute is used to determine whether the controller scales position, velocity, and acceleration attributes based on rotary or linear scaling parameters and their associated Drive Units that are defined by the SERCOS Interface standard. When the bit is clear (default), the corresponding bits in the SERCOS Position Data Scaling, Velocity Data Scaling, and Acceleration Data Scaling parameters are also cleared, which instructs the drive to use the rotary scaling parameters. When the bit is set, the corresponding bits in the SERCOS Position Data Scaling, Velocity Data Scaling, and Acceleration Data Scaling parameters are also set, which instructs the drive to use the linear scaling parameters. ***Continued on next page***

Attribute	Axis Type	Data Type	Access	Description
Drive Scaling Bits (cont.)				**Linear Scaling Unit** When the Scaling Unit is set to linear, the Linear Scaling bit attribute is used to determine whether the controller scales position, velocity, and acceleration attributes based on Metric or English Drive Units as defined by the SERCOS Interface standard. When the bit is clear (default), the corresponding bits in the SERCOS Position Data Scaling, Velocity Data Scaling, and Acceleration Data Scaling parameters are also cleared, which instructs the drive to use the Metric scaling parameters. When the bit is set, the corresponding bits in the SERCOS Position Data Scaling, Velocity Data Scaling, and Acceleration Data Scaling parameters are also set, which instructs the drive to scale in English units. If the Scaling Unit is set to rotary, the Linear Scaling Unit bit has no affect. When interfacing to Rockwell SERCOS drive products, the Standard Drive Units based on the Scaling Unit and Linear Scaling Unit bit selections are shown in the following table:

Standard Drive Units		
	Metric	**English**
Rotary	Rev	Rev
Linear	Millimeter	Inch

Data Reference

The Data Reference bit determines which side of the mechanical transmission to reference position, velocity, acceleration, and torque data. If motor is selected then position, velocity, acceleration, and torque data is referenced to the motor side of the transmission. If load is selected then position, velocity, acceleration, and torque data is referenced to the load-side of the transmission. This is only applicable when using an auxiliary feedback device.

Attribute	Axis Type	Data Type	Access	Description
Drive Status Bits	AXIS_SERVO_DRIVE	DINT	GSV	Lets you access all the drive status bits in one 32-bit word. This attribute is the same as the Drive Status tag.

Tag	Bit
Servo Action Status	0
Drive Enable Status	1
Shutdown Status	2
Process Status	3
Bus Ready Status	4
Reserved	5
Home Input Status	6
Reg 1 Input Status	7
Reg 2 Input Status	8
Pos Overtravel Input Status	9
Neg Overtravel Input Status	10
Enable Input Status	11
Accel Limit Status	12
Absolute Reference Status	13
Reserved	14
Reserved	15
Velocity Lock Status	16
Velocity Standstill Status	17
Velocity Threshold Status	18
Torque Threshold Status	19
Torque Limit Status	20
Velocity Limit Status	21
Position Lock Status	22
Power Limit Status	23
Reserved	24
Low Velocity Threshold Status	25
High Velocity Threshold Status	26

Attribute	Axis Type	Data Type	Access	Description
Drive Status	AXIS_SERVO_DRIVE	DINT	Tag	Lets you access all the drive status bits in one 32-bit word. This tag is the same as the Drive Status Bits attribute.

Tag	Bit
Servo Action Status	0
Drive Enable Status	1
Shutdown Status	2
Process Status	3
Bus Ready Status	4
Reserved	5
Home Input Status	6
Reg 1 Input Status	7
Reg 2 Input Status	8
Pos Overtravel Input Status	9
Neg Overtravel Input Status	10
Enable Input Status	11
Accel Limit Status	12
Absolute Reference Status	13
Reserved	14
Reserved	15
Velocity Lock Status	16
Velocity Standstill Status	17
Velocity Threshold Status	18
Torque Threshold Status	19
Torque Limit Status	20
Velocity Limit Status	21
Position Lock Status	22
Power Limit Status	23
Reserved	24
Low Velocity Threshold Status	25
High Velocity Threshold Status	26

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Drive Status</td><td>AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>
<table>
<tr><th>Tag</th><th>Bit</th></tr>
<tr><td>Servo Action Status</td><td>0</td></tr>
<tr><td>Drive Enable Status</td><td>1</td></tr>
<tr><td>Shutdown Status</td><td>2</td></tr>
<tr><td>Process Status</td><td>3</td></tr>
<tr><td>Bus Ready Status</td><td>4</td></tr>
<tr><td>Reserved</td><td>5</td></tr>
<tr><td>Home Input Status</td><td>6</td></tr>
<tr><td>Reg 1 Input Status</td><td>7</td></tr>
<tr><td>Reg 2 Input Status</td><td>8</td></tr>
<tr><td>Pos Overtravel Input Status</td><td>9</td></tr>
<tr><td>Neg Overtravel Input Status</td><td>10</td></tr>
<tr><td>Enable Input Status</td><td>11</td></tr>
<tr><td>Accel Limit Status</td><td>12</td></tr>
<tr><td>Absolute Reference Status</td><td>13</td></tr>
<tr><td>Safe-Off Mode Active Status</td><td>14 (requires Drive firmware revision 1.85 or higher)</td></tr>
</table>
</td></tr>
<tr><td>Drive Thermal Fault Action</td><td>AXIS_SERVO_DRIVE</td><td>SINT</td><td>GSV
SSV</td><td>
<table>
<tr><th>Fault Action</th><th>Value</th></tr>
<tr><td>Shutdown</td><td>0</td></tr>
<tr><td>Disable Drive</td><td>1</td></tr>
<tr><td>Stop Motion</td><td>2</td></tr>
<tr><td>Status Only</td><td>3</td></tr>
</table>
</td></tr>
<tr><td>Drive Undervoltage Fault</td><td>AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>Set when drive DC bus voltage is below the predefined operating limits for the bus.</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Drive Unit	AXIS_SERVO_DRIVE	INT	GSV	The Drive Unit attribute establishes the unit of measure that is applied to the Drive Resolution attribute value. Units appearing in the enumerated list may be linear or rotary, english or metric. Further discrimination is provided in the enumerated list to specify whether the Drive Unit is referenced directly to the motor or to the external, or auxiliary feedback. 0 = motor revs 1 = aux revs 2 = motor inches 3 = aux inches 4 = motor mm 5 = aux mm

Attribute	Axis Type	Data Type	Access	Description
Drive Warning Bits	AXIS_SERVO_DRIVE	DINT	GSV*	

Warning	Bit
Drive Overload Warning	0
Drive Overtemperature Warning	1
Motor Overtemperature Warning	2
Cooling Error Warning	3

Drive Overload Warning

When the load limit of the motor is exceeded, the Overload Warning bit is set. If the condition persists, an Overload Fault occurs. This warning bit gives the control program an opportunity to reduce motor loading to avoid a future shutdown situation.

Drive Overtemperature Warning

When the over-temperature limit of the drive is exceeded, the Drive Overtemperature Warning bit is set. If the condition persists, a Drive Overtemperature Fault occurs. This warning bit gives the control program an opportunity to reduce motor loading, or increasing drive cooling, to avoid a future shutdown situation.

Motor Overtemperature Warning

When the over-temperature limit of the motor is exceeded, the Motor Overtemperature Warning bit is set. If the condition persists, a Motor Overtemperature Fault occurs. This warning bit gives the control program an opportunity to reduce motor loading, or increasing motor cooling, to avoid a future shutdown situation.

Cooling Error Warning

When the ambient temperature limit inside the drive enclosure is exceeded, the Cooling Error Warning bit sets. If the condition persists, a Cooling Error Fault occurs. This warning bit gives the control program an opportunity to increase drive cooling to avoid a future shutdown situation.

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Dynamics Configuration Bits</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>DINT</td><td>GSV
SSV</td><td>Revision 16 improved how the controller handles changes to an S-curve profile.

Do you want to return to revision 15 or earlier behavior for S-curves?

· NO — Leave these bits ON (default).
· YES — Turn OFF one or more of these bits.

<table>
<tr><th>To turn off this change</th><th>Turn off this bit</th></tr>
<tr><td>Reduced S-curve Stop Delay

This change applies to the Motion Axis Stop (MAS) instruction. It lets you use a higher deceleration jerk to stop an accelerating axis more quickly.

The controller uses the deceleration jerk of the stopping instruction if it is more than the current acceleration jerk.</td><td>0</td></tr>
<tr><td>Reduced S-curve Velocity Reversals

Before revision 16, you could cause an axis to momentarily reverse direction if you decreased the deceleration jerk while the axis was decelerating. This typically happened if you tried to restart a jog or move with a lower deceleration rate while the axis was stopping. This change prevents the axis from reversing in those situations.</td><td>1</td></tr>
<tr><td>Reduced S-curve Velocity Overshoots

You can cause an axis to overshoot its programmed speed if you decrease the acceleration jerk while the axis is accelerating. This change keeps to overshoot to no more than 50% of the programmed speed.</td><td>2</td></tr>
</table>
</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Enable Input Status	AXIS_SERVO_DRIVE	BOOL	Tag	If this bit is: · ON — The Enable input is active. · OFF — The Enable input is inactive.
External Drive Type	AXIS_SERVO_DRIVE	DINT	GSV SSV	0 = torque servo 1 = velocity servo 2 = hydraulic servo When the application requires the servo module axis to interface with an external velocity servo drive, the External Drive Type should be configured for velocity servo. This disables the servo module's internal digital velocity loop. If the External Drive Type attribute is set to torque servo, the servo module's internal digital velocity loop is active. This configuration is the required configuration for interfacing to a torque loop servo drive. If the External Drive Type attribute is set to hydraulic servo, the object will enable certain features specific to hydraulic servo applications. In general, selecting the hydraulic External Drive Type configures the servo loop the same as selecting the velocity servo External Drive Type.

Attribute	Axis Type	Data Type	Access	Description
Fault Configuration Bits	AXIS_SERVO AXIS_SERVO_DRIVE	DINT	GSV SSV	

Axis Type	Fault Configuration	Bit
AXIS_SERVO	Soft Overtravel Checking	0
	Reserved	1
	Drive Fault Checking	2
	Drive Fault Normally Closed	3
AXIS_SERVO_DRIVE	Soft Overtravel Checking	0
	Hard Overtravel Checking	1
	Reserved	2
	Reserved	3
	Drive Enable Input Fault Handling	4
	Drive Enable Input Checking	5

Change to rotary or Overtravel Checking requires Home range checks.

Soft Overtravel Checking

Soft overtravel checking is only available for a linear axis.

Do you want a Positive Soft Overtravel Fault or Negative Soft Overtravel Fault to happen if the axis goes outside the configured travel limits?

- YES — Set this bit.
- NO — Clear this bit.

The Maximum Positive Travel and Maximum Negative Travel attributes set the travel limits. This check supplements but doesn't replace hardware overtravel fault protection that uses hardware limit switches to directly stop axis motion at the drive and deactivate power to the system.

Hard Overtravel Checking

Hard overtravel checking is only available for a linear axis.

Do you want a Positive Hard Overtravel Fault or Negative Hard Overtravel Fault to happen if the axis activates the positive or negative overtravel limit switch inputs?

- YES — Set this bit.
- NO — Clear this bit.

Drive Fault Checking

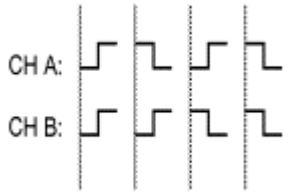
The motion module provides a dedicated drive fault input for each axis. These inputs may be connected to fault outputs on the external drive (if provided) to notify the servo module of a fault in the drive itself. Set the Drive Fault Checking bit if you are using the servo module's drive fault input, and then specify the drive fault contact configuration of the amplifier's drive fault output as described below.

Continued on next page

Attribute	Axis Type	Data Type	Access	Description
Fault Configuration Bits (cont.)		DINT	GSV SSV	**Drive Fault Normally Closed** The Drive Fault Normally Closed bit attribute controls the sense of the Drive Fault input to the servo module. If this bit is set (true) then during normal (fault-free) operation of the drive, the Drive Fault input should be active, that is, 24 Volts. If a drive fault occurs, the drive will open its drive fault output contacts and remove 24 Volts from the servo module's Drive Fault input generating an axis Drive Fault condition. This is the default "fail-safe" configuration. In some cases it may be necessary to clear the Drive Fault Normally Closed bit to interface with a drive system that closes its contacts when faulted. This is generally not recommended for "fail-safe" operation. **Drive Enable Input Fault Handling** When the Drive Enable Input Fault Handling bit is set, it lets the drive post a fault based on the condition of the Drive Enable Input. If an attempt is made to enable the drive axis without an active Drive Enable Input, the drive sets a Drive Enable Input Fault. If the Drive Enable Input ever goes from active to inactive while the drive axis is enabled, the drive also sets a Drive Enable Input Fault. If the Drive Enable Input Fault Handling bit is clear (default), then the drive does not generate a Drive Enable Input Fault. **Drive Enable Input Checking** When the Drive Enable Input Checking bit is set (the default) the drive regularly checks the current state of the Drive Enable Input. This dedicated input serves as a permissive to enable the drive's power structure and servo loop. Once the drive is enabled, a transition of the Drive Enable Input from active to inactive results in a drive initiated axis stop where the axis is decelerated to a stop using the configured Stopping Torque and then disabled. If the drive enable Input Checking bit is clear, then no Drive Enable Input checking is done, hence the state of the input is irrelevant to drive operation. The state of the switch is still reported as part of the Drive Status bits attribute.

Attribute	Axis Type	Data Type	Access	Description
Feedback Fault	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	**AXIS_SERVO** Set for a specific feedback source when one of the following conditions occurs: · The differential electrical signals for one or more of the feedback channels (for example, A+ and A-, B+ and B-, or Z+ and Z-) are at the same level (both high or both low). Under normal operation, the differential signals are always at opposite levels. The most common cause of this situation is a broken wire between the feedback transducer and the servo module or drive; · Loss of feedback "power" or feedback "common" electrical connection between the servo module or drive and the feedback device. The controller latches this fault. Use a Motion Axis Fault Reset (MAFR) or Motion Axis Shutdown Reset (MASR) instruction to clear the fault. **AXIS_SERVO_DRIVE** Set when one of the feedback sources associated with the drive axis has a problem that prevents the drive from receiving accurate or reliable position information from the feedback device. Set when one of the feedback sources for the axis can't send accurate or reliable position information because there is a problem. For AXIS_SERVO axis, possible problems are: · The differential electrical signals for one or more of the feedback channels (for example, A+ and A-, B+ and B-, or Z+ and Z-) are at the same level (both high or both low). Under normal operation, the differential signals are always at opposite levels. The most common cause of this situation is a broken wire between the feedback transducer and the servo module or drive; · Loss of feedback power or common electrical connection between the servo module or drive and the feedback device. The controller latches this fault. Use a Motion Axis Fault Reset (MAFR) or Motion Axis Shutdown Reset (MASR) instruction to clear the fault.
Feedback Fault Action	AXIS_SERVO AXIS_SERVO_DRIVE	SINT	GSV SSV	

Fault Action	Value
Shutdown	0
Disable Drive	1
Stop Motion	2
Status Only	3

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Feedback Noise Fault</td><td>AXIS_SERVO</td><td>BOOL</td><td>Tag</td><td>Set when there is noise on the feedback device's signal lines.
· For example, simultaneous transitions of the feedback A and B channels of an A Quad B is referred to generally as feedback noise.
· Feedback noise (shown below) is most often caused by loss of quadrature in the feedback device itself or radiated common-mode noise signals being picked up by the feedback device wiring. You can see both of these on an oscilloscope.
 

· To troubleshoot the loss of channel quadrature, look for:
▪ physical misalignment of the feedback transducer components
▪ excessive capacitance (or other delays) on the encoder signals
· Proper grounding and shielding usually cures radiated noise problems.
The controller latches this fault. Use a Motion Axis Fault Reset (MAFR) or Motion Axis Shutdown Reset (MASR) instruction to clear the fault.</td></tr>
<tr><td>Feedback Noise Fault Action</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>SINT</td><td>GSV
SSV</td><td><table><tr><th>Fault Action</th><th>Value</th></tr><tr><td>Shutdown</td><td>0</td></tr><tr><td>Disable Drive</td><td>1</td></tr><tr><td>Stop Motion</td><td>2</td></tr><tr><td>Status Only</td><td>3</td></tr></table></td></tr>
<tr><td>Friction Compensation</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
SSV</td><td>0…100%

It is not unusual for an axis to have enough static friction (sticktion) that even with a significant position error it won't move. Integral gain can be used to generate enough output to the drive to correct the error, but this approach may not be responsive enough for the application. An alternative is to use Friction Compensation to break sticktion in the presence of a non-zero position error. This is done by adding, or subtracting, a fixed output level, called Friction Compensation, to the Servo Output value based on its current sign.

The Friction Compensation value should be just under the value that would break the sticktion. A larger value causes the axis to dither. Dither is when the axis moves rapidly back and forth centered on the commanded position.</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Friction Compensation Window	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units To address the issue of dither when applying Friction Compensation and hunting from the integral gain, a Friction Compensation Window is applied around the current command position when the axis is not being commanded to move. If the actual position is within the Friction Compensation Window the Friction Compensation value is applied to the Servo Output but scaled by the ratio of the position error to the Friction Compensation Window. Within the window, the servo integrators are also disabled. Thus, once the position error reaches or exceeds the value of the Friction Compensation Window attribute, the full Friction Compensation value is applied. Of course, should the Friction Compensation Window be set to zero, this feature is effectively disabled. A non-zero Friction Compensation Window has the effect of softening the Friction Compensation as its applied to the Servo Output and reducing the dithering effect that it can create. This generally allows higher values of Friction Compensation to be applied. Hunting is also eliminated at the cost of a small steady-state error.

Attribute	Axis Type	Data Type	Access	Description
Guard Status	AXIS_SERVO_DRIVES	DINT	Tag GSV	

Tag	Bit
Guard OK	0
Guard Config Locked	1
Guard Gate Drive Output	2
Guard Stop Input	3
Guard Stop Request	4
Guard Stop in Progress	5
Guard Stop Decel	6
Guard Stop Standstill	7
Guard Stop Output	8
Guard Limited Speed Input	9
Guard Limited Speed Request	10
Guard Limited Speed Monitor in Progress	11
Guard Limited Speed Output	12
Guard Max Speed Monitor in Progress	13
Guard Max Accel Monitor in Progress	14
Guard Direction Monitor in Progress	15
Guard Door Control Lock	16
Guard Door Control Output	17
Guard Door Monitor Input	18
Guard Door Monitor In Progress	19
Guard Lock Monitor Input	20
Guard Enabling Switch Input	21
Guard Enabling Switch in Progress	22
Guard Reset Input	23
Guard Reset Required	24
Guard Stop Input Cycle Required	25
Reserved	26
Reserved	27
Reserved	28
Reserved	29
Reserved	30
Reserved	31

Attribute	Axis Type	Data Type	Access	Description
Guard Faults	AXIS_SERVO_DRIVES	DINT	Tag GSV	

Tag	Bit
Reserved	0
Guard Internal Fault	1
Guard Configuration Fault	2
Guard Gate Drive Fault	3
Guard Reset Fault	4
Guard Feedback 1 Fault	5
Guard Feedback 2 Fault	6
Guard Feedback Speed Compare Fault	7
Guard Feedback Position Compare Fault	8
Guard Stop Input Fault	9
Guard Stop Output Fault	10
Guard Stop Decel Fault	11
Guard Stop Standstill Fault	12
Guard Stop Motion Fault	13
Guard Limited Speed Input Fault	14
Guard Limited Speed Output Fault	15
Guard Limited Speed Monitor Fault	16
Guard Max Speed Monitor Fault	17
Guard Max Accel Monitor Fault	18
Guard Direction Monitor Fault	19
Guard Door Monitor Input Fault	20
Guard Door Monitor Fault	21
Guard Door Control Output Fault	22
Guard Lock Monitor Input Fault	23
Guard Lock Monitor Fault	24
Guard Enabling Switch Monitor Input Fault	25
Guard Enabling Switch Monitor Fault	26
Guard Feedback 1 Voltage Monitor Fault	27
Guard Feedback 2 Voltage Monitor Fault	28
Reserved	29
Reserved	30
Reserved	31

Do you want any of these faults to give the controller a major fault?

- YES - Set the General Fault Type of the motion group = Major Fault.
- NO - You must write code to handle these faults.

For more information about error codes displayed on drives and/or multi-axis motion control systems, see page 385.

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Gearing Lock Status</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>BOOL</td><td>Tag</td><td>Set whenever the slave axis is locked to the master axis in a gearing relationship according to the specified gear ratio. The clutch function of the gearing planner is used to ramp an axis up, or down, to speed in a gearing process (MAG with Clutch selected). This bit is cleared during the intervals where the axis is clutching.</td></tr>
<tr><td>Gearing Status</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>BOOL</td><td>Tag</td><td>Set if the axis is a slave that is currently gearing to another axis. Cleared when the gearing operation is stopped or is superseded by some other motion operation.</td></tr>
<tr><td>Ground Short Fault</td><td>AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>When the drive detects an imbalance in the DC bus supply current, the Ground Short Fault bit is set, indicating that current is flowing through an improper ground connection.</td></tr>
<tr><td>Group Instance</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>DINT</td><td>GSV</td><td>Instance Number of Group assigned to Axis

The Group Instance attribute is used to determine what motion group object instance this axis is assigned to.</td></tr>
<tr><td>Hard Overtravel Fault Action</td><td>AXIS_SERVO_DRIVE</td><td>SINT</td><td>GSV
SSV</td><td><table><tr><th>Fault Action</th><th>Value</th></tr><tr><td>Shutdown</td><td>0</td></tr><tr><td>Disable Drive</td><td>1</td></tr><tr><td>Stop Motion</td><td>2</td></tr><tr><td>Status Only</td><td>3</td></tr></table></td></tr>
<tr><td>Home Configuration Bits</td><td>AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>DINT</td><td>GSV
SSV</td><td>0 = (Reserved)

1 = Home Switch Normally Closed

2 = Marker Edge Negative

Home Switch Normally Closed

The Home Switch Normally Closed bit attribute determines the normal state of the home limit switch used by the homing sequence. The normal state of the switch is its state prior to being engaged by the axis during the homing sequence. For example, if the Home Switch Normally Closed bit is set (true) then the condition of the switch prior to homing is closed. When the switch is engaged by the axis during the homing sequence, the switch is opened, which constitutes a homing event.</td></tr>
<tr><td>Home Direction</td><td>AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>SINT</td><td>GSV
SSV</td><td>0 = unidirectional forward

1 = bidirectional forward

2 = unidirectional reverse

3 = bidirectional reverse</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Home Event Armed Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set when a home event has been armed through execution of the MAH (Motion Axis Home) instruction. Cleared when a home event occurs.
Home Event Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set when a home event has occurred. Cleared when another MAH (Motion Axis Home) instruction is executed.
Home Event Task	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	MSG	User Event Task that is triggered to execute when a Home event occurs. An instance value of 0 indicates that no event task has been configured to be triggered by the Home Event. This attribute indicates which user Task is triggered when a home event occurs. The user Task is triggered at the same time that the Process Complete bit is set for the instruction that armed the home event. This attribute is set through internal communication from the user Task object to the Axis object when the Task trigger attribute is set to select the Home Event Task Instance attribute of the Axis. This attribute should not be set directly by an external device. This attribute is available to be read externally (Get attributes List) for diagnostic information.
Home Input Status	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	If this bit is: · ON — The home input is active. · OFF — The home input is inactive.
Home Mode	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	SINT	GSV SSV	0 = passive 1 = active (default) 2 = absolute
Home Offset	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV SSV	Position Units When applied to an active or passive Homing Mode, using a non-immediate Home Sequence, the Home Offset is the desired position offset of the axis Home Position from the position at which the home event occurred. The Home Offset is applied at the end of the specified homing sequence before the axis moves to the Home Position. In most cases, Home Offset is set to zero. After an active bidirectional homing sequence has completed, the axis is left at the specified Home Position. If the Home Offset is non-zero, the axis will then be offset from the marker or home switch event point by the Home Offset value. If the Home Offset is zero, the axis will sit right “on top of” the marker or home switch point.

Attribute	Axis Type	Data Type	Access	Description
Home Position	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV SSV	Position Units The Home Position is the desired absolute position for the axis after the specified homing sequence has been completed. After an active homing sequence has completed, the axis is left at the specified Home Position. In most cases, Home Position is set to zero, although any value, within the Maximum Positive and Negative Travel limits of the axis (if enabled), may also be used. (A description of the Maximum Positive and Negative Travel configuration attributes may be found in the Servo and Drive Axis Object specifications). For a rotary axis, the Home Position is constrained to be a positive number less than the Position Unwind value divided by the Conversion Constant. When configured for absolute Homing Mode, the Home Position value is applied directly to the absolute feedback device to establish an absolute position reference for the system.
Home Return Speed	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units / Sec The Home Return Speed attribute controls the speed of the jog profile used after the first leg of an active bidirectional homing sequence.
Home Sequence	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	SINT	GSV SSV	0 = immediate (default) 1 = switch 2 = marker 3 = switch then marker 4 = torque limit 5 = torque limit then marker
Home Speed	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units / Sec The Home Speed attribute controls the speed of the jog profile used in the first leg of an active homing sequence as described in the above discussion of the Home Sequence Type attribute.
Homed Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Cleared at power-up or reconnection. Set by the MAH instruction upon successful completion of the configured homing sequence, and later cleared when the axis enters the shutdown state.
Homing Status	AXIS_CONSUMED AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set if a Home motion profile is currently in progress. Cleared when the homing operation is stopped or is superseded by some other motion operation.

Attribute	Axis Type	Data Type	Access	Description
Inhibit Status	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	Use the InhibitStatus bit of an axis to see if the axis is inhibited or uninhibited. If the bit is: · ON — The axis is inhibited. · OFF — The axis is uninhibited. The controller changes the InhibitStatus bit only after all of these have happened: · The axis has changed to inhibited or uninhibited. · All uninhibited axes are ready. · The connections to the motion module are running again.
InhibitAxis	AXIS_SERVO AXIS_SERVO_DRIVE	INT	GSV SSV	**To** / **Set the attribute to** Block the controller from using the axis. This inhibits the axis. / 1 or any non-zero value Let the controller use the axis. This uninhibits the axis. / 0
Integrator Hold Enable	AXIS_SERVO AXIS_SERVO_DRIVE	SINT	GSV SSV	When the Integrator Hold Enable attribute value is configured TRUE, the servo loop temporarily disables any enabled integrators while the command position is changing. This feature is used by point-to-point moves to minimize the integrator wind-up during motion. When the Integrator Hold Enable attribute value is FALSE, all active integrators are always enabled. 0 = disabled 1 = enabled
Inter Module Sync Fault	AXIS_SERVO	BOOL	Tag	If this bit is on, the analog servo cards of a SoftLogix5800 controller aren't synchronized. The hardware or vbfirmware of the card causes this fault. For example, the cable between 2 cards isn't connected.
Interpolated Actual Position	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	Interpolated Actual Position in Position Units Interpolated Actual Position is the interpolation of the actual position, based on past axis trajectory history, at the time specified by the "Interpolated Time" attribute.
Interpolated Command Position	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	Interpolated Command Position in Position Units Interpolated Command Position is the interpolation of the commanded position, based on past axis trajectory history, at the time specified by the "Interpolated Time" attribute.
Interpolation Time	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	GSV Tag	CST time to interpolate to Interpolated Time is the 32-bit CST time used to calculate the interpolated positions. When this attribute is updated with a valid CST value, the Interpolated Actual Position and Interpolated Command Position values are automatically calculated.

Attribute	Axis Type	Data Type	Access	Description
Jog Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set if a Jog motion profile is currently in progress. Cleared when the Jog is complete or is superseded by some other motion operation.
LDT Calibration Constant	AXIS_SERVO	REAL	GSV	This attribute provides for setting a calibration constant for LDT devices. This attribute is only active if the Transducer Type is set to LDT.
LDT Calibration Constant Units	AXIS_SERVO	SINT	GSV	0 = m/sec 1 = Usec/in This attribute provides a selection for the units of the LDT calibration constant attribute. This attribute is only active if the Transducer Type is set to LDT.
LDT Length	AXIS_SERVO	REAL	GSV	This attribute provides for setting the length of an LDT device. This attribute is only active if the Transducer Type is set to LDT.
LDT Length Units	AXIS_SERVO	SINT	GSV	0 = m 1 = in This attribute provides a selection for the units of the LDT length attribute. This attribute is only active if the Transducer Type is set to LDT.
LDT Recirculations	AXIS_SERVO	SINT	GSV	This attribute provides the number of recirculations. This attribute is only active if the Transducer Type is set to LDT and LDT Type is set to PWM.
LDT Scaling	AXIS_SERVO	REAL	GSV	This attribute provides for setting the scaling factor for LDT devices. This attribute is only active if the Transducer Type is set to LDT.
LDT Scaling Units	AXIS_SERVO	SINT	GSV	0 = Position Units/m 1 = Position Units/in This attribute provides a selection for the units of the LDT scaling attribute. This attribute is only active if the Transducer Type is set to LDT.
LDT Type	AXIS_SERVO	SINT	GSV	0 = PWM 1 = Start/Stop Rising 2 = Start/Stop Falling This attribute provides a selection for the LDT Type. It provides the following enumerated values: PWM, Start/Stop Rising, and Start/Stop Falling. This attribute is only active if the Transducer Type is set to LDT.

Attribute	Axis Type	Data Type	Access	Description
Load Inertia Ratio	AXIS_SERVO_DRIVE	REAL	GSV SSV	%Rated / Pos Units per Sec^2 The Motor Inertia value represents the inertia of the motor without any load attached to the motor shaft in Torque Scaling units of %Rated / Pos Units per Sec^2. The Load Inertia Ratio attribute's value represents the ratio of the load inertia to the motor inertia. Auto-tuning uses the Motor Inertia value to calculate the Load Inertia Ratio based on the following equation. Load Inertia Ratio = (Total Inertia - Motor Inertia) / Motor Inertia. Total Inertia is directly measured by the auto-tuning algorithm and applied to the Torque Scaling attribute in units of %Rated / Pos Units per Sec^2. If the Load Inertia Ratio value is known, the Motor Inertia value can also be used to calculate a suitable Torque Scaling value for the fully loaded motor without performing an auto-tune. The equation used by RSLogix5000 to calculate the Torque Scaling value is as follows: Torque Scaling = (1 + Load Inertia Ratio) * Motor Inertia. The value for Load Inertia may be automatically calculated using Rockwell's MotionBook program while the value for Motor Inertia is derived from the Motion database file based on the motor selection.
Map Instance	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	DINT	GSV	I/O Map Instance Number. This is 0 for virtual and consumed Data Types. The axis is associated to a specific motion compatible module by specifying the instance of the map entry representing the module.
Marker Distance	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important**: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. Marker Distance in Position Units Marker Distance is the distance between the axis position at which a home switch input was detected and the axis position at which the marker event was detected. This value is useful in aligning a home limit switch relative to a feedback marker pulse to provide repeatable homing operation.

Attribute	Axis Type	Data Type	Access	Description
Master Input Configuration Bits	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	GSV SSV	Bits 0 = Master Delay Compensation 1 = Master Position Filter **Master Delay Compensation** By default, both the Position Camming and Gearing functions, when applied to a slave axis, perform Master Delay Compensation to compensate for the delay time between reading the master axis command position and applying the associated slave command position to the input of the slave's servo loop. When the master axis is running at a fixed speed, this compensation technique insures that the slave axis command position accurately tracks the actual position of the master axis; in other words, Master Delay Compensation allows for zero tracking error when gearing or camming to the actual position of a master axis. The Master Delay Compensation algorithm extrapolates the position of the master axis at the predicted time when the command position is applied to the slave's servo loop. Since master axis position is measured in discrete feedback counts and is inherently noisy, the extrapolation process amplifies that noise according to the total position update delay. The total position update delay is proportional to the Coarse Update Period of the motion group, and, if the master or the slave involves an AXIS_SERVO_DRIVE data type, it also includes the delay term that is proportional to the SERCOS Update Period. The greater the delay, the greater the noise introduced by the extrapolator. The Master Delay Compensation feature also has an extrapolation filter to filter the noise introduced by the extrapolation process. The time constant of the filter is fixed at 4x the total position update delay (independent of the Master Position Filter Bandwidth), which again is a function of the Coarse Update Period (and the SERCOS Update Period, if a AXIS_SERVO_DRIVE data type). The controller uses a 1st order extrapolation algorithm that results in zero tracking error while the master axis is moving at constant velocity. If the master axis accelerates or decelerates the tracking error is non-zero and proportional to the acceleration or deceleration rate and also proportional to the square of the total position update delay time. From both a noise and acceleration error perspective, minimizing the coarse update period is vital. Some applications don't need zero tracking error between the master and the slave axis. In these cases, it may be beneficial to disable the Master Delay Compensation feature to eliminate the disturbances the extrapolation algorithm introduces to the slave axis. When the Master Delay Compensation feature is disabled (bit cleared), the slave axis will appear to be more responsive to movements of the master and run generally smoother than when Master Delay Compensation feature is enabled (bit set). However, when the master axis is running at a constant velocity, the slave will lag the master by a tracking error that is proportional to the speed of the master. Note that Master Delay Compensation, even if explicitly enabled, is not applied in cases where a slave axis is gearing or camming to the master axis' command position. Since the controller generates the command position directly, there is no intrinsic master position delay to compensate for.

Continued on next page

Attribute	Axis Type	Data Type	Access	Description
Master Input Configuration Bits (cont.)				**Master Position Filter** The Master Position Filter bit controls the activity of an independent single-pole low-pass filter that effectively filters the specified master axis position input to the slave's gearing or position camming operation. When enabled (bit set), this filter has the effect of smoothing out the actual position signal from the master axis, and thus smoothing out the corresponding motion of the slave axis. The trade-off for smoothness is an increase in lag time between the response of the slave axis to changes in motion of the master. Note that the Master Position Filter also provides filtering to the extrapolation noise introduced by the Master Delay Compensation algorithm, if enabled. When the Master Position Filter bit is set, the bandwidth of the Master Position Filter is controlled by the Master Position Filter Bandwidth attribute, see below. This can be done by setting the Master Position Filter bit and controlling the Master Position Filter Bandwidth directly. Setting the Master Position Filter Bandwidth to zero can be used to effectively disable the filter.
Master Offset	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	**Important**: To use this attribute, make sure Auto Tag Update is Enabled for the motion group (default setting). Otherwise, you won't see the right value as the axis runs. Master Offset in Master Position Units The Master Offset is the position offset that is currently applied to the master side of the position cam. The Master Offset is returned in master position units. The Master Offset will show the same unwind characteristic as the position of a linear axis.
Master Offset Move Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set if a Master Offset Move motion profile is currently in progress. This bit is cleared when the Master Offset Move is complete or is superseded by some other motion operation.
Master Position Filter Bandwidth	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV SSV	Hertz The Master Position Filter Bandwidth attribute controls the activity of the single-pole low-pass filter that filters the specified master axis position input to the slave's gearing or position camming operation. When enabled, this filter has the effect of smoothing out the actual position signal from the master axis, and thus smoothing out the corresponding motion of the slave axis. The trade-off for smoothness is an increase in lag time between the response of the slave axis to changes in motion of the master. If the Master Position Filter is disabled, the Master Position Filter Bandwidth has no effect.

Attribute	Axis Type	Data Type	Access	Description
Maximum Acceleration	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV SSV	Position Units / Sec^2 The Maximum Acceleration and Deceleration attribute values are frequently used by motion instructions such as MAJ, MAM, MCD, and so on, to determine the acceleration and deceleration rates to apply to the axis. These instructions all have the option of specifying acceleration and deceleration as a percent of the Maximum Acceleration and Maximum Deceleration attributes for the axis. The Maximum Acceleration and Maximum Deceleration values for the axis are automatically set to ~ 85% of the measured Tune Acceleration and Tune Deceleration by the MAAT (Motion Apply Axis Tune) instruction. If set manually, these values should typically be set to ~85% of the maximum acceleration and maximum deceleration rate of the axis. This provides sufficient "head-room" for the axis to operate at all times within the acceleration and deceleration limits of the drive and motor.
Maximum Deceleration	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV SSV	Position Units / Sec^2 The Maximum Acceleration and Deceleration attribute values are frequently used by motion instructions such as MAJ, MAM, MCD, and so on, to determine the acceleration and deceleration rates to apply to the axis. These instructions all have the option of specifying acceleration and deceleration as a percent of the Maximum Acceleration and Maximum Deceleration attributes for the axis. The Maximum Acceleration and Maximum Deceleration values for the axis are automatically set to ~ 85% of the measured Tune Acceleration and Tune Deceleration by the MAAT (Motion Apply Axis Tune) instruction. If set manually, these values should typically be set to ~85% of the maximum acceleration and maximum deceleration rate of the axis. This provides sufficient "head-room" for the axis to operate at all times within the acceleration and deceleration limits of the drive and motor.
Maximum Negative Travel	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units The Axis Object provides configurable software travel limits via the Maximum Positive and Negative Travel attributes. If the axis is configured for software overtravel limit checking by setting the Soft Overtravel Bit and the axis passes outside these maximum travel limits, a Software Overtravel Fault is issued. When software overtravel checking is enabled, appropriate values for the maximum travel in both the Maximum Positive and Maximum Negative Travel attributes need to be established with Maximum Positive Travel always greater than Maximum Negative Travel. Both of these values are specified in the configured Position Units of the axis. Note: The software travel limits are not enabled until the selected homing sequence is completed.

Attribute	Axis Type	Data Type	Access	Description
Maximum Positive Travel	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units The Axis Object provides configurable software travel limits via the Maximum Positive and Negative Travel attributes. If the axis is configured for software overtravel limit checking by setting the Soft Overtravel Bit and the axis passes outside these maximum travel limits, a Software Overtravel Fault is issued. When software overtravel checking is enabled, appropriate values for the maximum travel in both the Maximum Positive and Maximum Negative Travel attributes need to be established with Maximum Positive Travel always greater than Maximum Negative Travel. Both of these values are specified in the configured Position Units of the axis. Note: The software travel limits are not enabled until the selected homing sequence is completed.
Maximum Speed	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV SSV	Position Units / Sec The value of the Maximum Speed attribute is used by various motion instructions (for example, MAJ, MAM, MCD, and so on) to determine the steady-state speed of the axis. These instructions all have the option of specifying speed as a percent of the Maximum Speed attribute value for the axis. The Maximum Speed value for the axis is automatically set to the Tuning Speed by the MAAT (Motion Apply Axis Tune) instruction. This value is typically set to ~90% of the maximum speed rating of the motor. This provides sufficient "head-room" for the axis to operate at all times within the speed limitations of the motor.
Memory Usage	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	MSG	Amount of memory consumed for this instance (in bytes)
Memory Use	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	INT	GSV	Controller memory space where instance exists. 105 (0x69) = I/O space 106 (0x6a) = Data Table space RSLogix 5000 software uses this attribute to create axis instances in I/O memory for axes that are either to be produced or consumed. The Memory Use attribute can only be set as part of an axis create service and is used to control which controller memory the object instance is created in.
Module Channel	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE	SINT	GSV	Zero based channel number of the module. 0xff, indicates unassigned. The axis is associated to a specific channel on a motion module by specifying the Module Channel attribute.

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Module Class Code</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>DINT</td><td>GSV</td><td>ASA Object class code of the motion engine in the module; for example, 0xAF for the M02AE module.

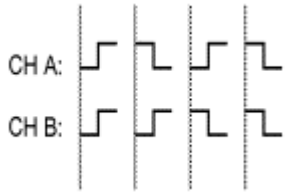
The ASA class code of the object in the motion module which is supporting motion; for example, 0xAF is the ASA object ID of the “Servo Module Axis Object” residing in the 1756-M02AE module.</td></tr>
<tr><td>Module Fault</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>BOOL</td><td>Tag</td><td>Set when a serious fault has occurred with the motion module associated with the selected axis. Usually a module fault affects all axes associated with the motion module. A module fault generally results in the shutdown of all associated axes. Reconfiguration of the motion module is required to recover from a module fault condition.

Do you want this fault to give the controller a major fault?
· YES — Set the General Fault Type of the motion group = Major Fault.
· NO — You must write code to handle these faults.</td></tr>
<tr><td>Module Fault Bits</td><td>AXIS_CONSUMED
AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>DINT</td><td>GSV*</td><td>Lets you access the module fault bits in one 32-bit word. This attribute is the same as the Module Faults tag.

<table><tr><th>Module Fault</th><th>Bit</th></tr><tr><td>Control Sync Fault</td><td>0</td></tr><tr><td>Module Sync Fault</td><td>1</td></tr><tr><td>Timer Event Fault</td><td>2</td></tr><tr><td>Module Hardware Fault</td><td>3</td></tr><tr><td>SERCOS Ring Fault</td><td>4</td></tr><tr><td>Inter Module Sync Fault</td><td>5</td></tr></table>
These faults have module scope instead of axis scope.
· These faults show up in all the axes that are connected to the motion module.
· The motion planner updates these fault bits every coarse update period.
Do you want any of these faults to give the controller a major fault?
· YES — Set the General Fault Type of the motion group = Major Fault.
· NO — You must write code to handle these faults.</td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Module Faults</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>DINT</td><td>Tag</td><td>Lets you access the module fault bits in one 32-bit word. This tag is the same as the Module Fault Bits attribute.

<table><tr><th>Module Fault</th><th>Bit</th></tr><tr><td>Control Sync Fault</td><td>0</td></tr><tr><td>Module Sync Fault</td><td>1</td></tr><tr><td>Timer Event Fault</td><td>2</td></tr><tr><td>Module Hardware Fault</td><td>3</td></tr><tr><td>SERCOS Ring Fault</td><td>4</td></tr><tr><td>Inter Module Sync Fault</td><td>5</td></tr></table>
These faults have module scope instead of axis scope.
· These faults show up in all the axes that are connected to the motion module.
· The motion planner updates these fault bits every coarse update period.
Do you want any of these faults to give the controller a major fault?
· YES — Set the General Fault Type of the motion group = Major Fault.
· NO — You must write code to handle these faults.</td></tr>
<tr><td>Module Hardware Fault</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>If this bit is set, the motion module has a hardware problem that, in general, is going to require replacement of the module.</td></tr>
<tr><td>Module Sync Fault</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>If this bit is set, the motion module lost communication with the controller and missed several position updates in a row.
· The motion module can miss up to 4 position updates. After that, the motion module shuts down.
· This bit clears when communication is reestablished.</td></tr>
<tr><td>Mot Feedback Fault</td><td>AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>Set for the A Quad B feedback device when one of these happens:
· The differential electrical signals for one or more of the feedback channels (for example, A+ and A-, B+ and B-, or Z+ and Z-) are at the same level (both high or both low). Under normal operation, the differential signals are always at opposite levels. The most common cause of this situation is a broken wire between the feedback transducer and the servo module or drive.
· Loss of feedback “power” or feedback “common” electrical connection between the servo module or drive and the feedback device.
The controller latches this fault. Use a Motion Axis Fault Reset (MAFR) or Motion Axis Shutdown Reset (MASR) instruction to clear the fault.</td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Mot Feedback Noise Fault</td><td>AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>Set when there is noise on the feedback device's signal lines.
· For example, simultaneous transitions of the feedback A and B channels of an A Quad B is referred to generally as feedback noise.
· Feedback noise (shown below) is most often caused by loss of quadrature in the feedback device itself or radiated common-mode noise signals being picked up by the feedback device wiring. You can see both of these on an oscilloscope.
 

· To troubleshoot the loss of channel quadrature, look for:
▪ physical misalignment of the feedback transducer components
▪ excessive capacitance (or other delays) on the encoder signals
· Proper grounding and shielding usually cures radiated noise problems.
The controller latches this fault. Use a Motion Axis Fault Reset (MAFR) or Motion Axis Shutdown Reset (MASR) instruction to clear the fault.</td></tr>
<tr><td>Motion Status</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>DINT</td><td>Tag</td><td>Lets you access all the motion status bits in one 32-bit word. This tag is the same as the Motion Status Bits attribute.
<table><tr><th>Motion Status</th><th>Bit</th></tr><tr><td>Accel Status</td><td>0</td></tr><tr><td>Decel Status</td><td>1</td></tr><tr><td>Move Status</td><td>2</td></tr><tr><td>Jog Status</td><td>3</td></tr><tr><td>Gearing Status</td><td>4</td></tr><tr><td>Homing Status</td><td>5</td></tr><tr><td>Stopping Status</td><td>6</td></tr><tr><td>Homed Status</td><td>7</td></tr><tr><td>Position Cam Status</td><td>8</td></tr><tr><td>Time Cam Status</td><td>9</td></tr><tr><td>Position Cam Pending Status</td><td>10</td></tr><tr><td>Time Cam Pending Status</td><td>11</td></tr><tr><td>Gearing Lock Status</td><td>12</td></tr><tr><td>Position Cam Lock Status</td><td>13</td></tr><tr><td>Reserved</td><td>14</td></tr><tr><td>Master Offset Move Status</td><td>15</td></tr><tr><td>Coordinated Motion Status</td><td>16</td></tr></table></td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Motion Status Bits</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>DINT</td><td>GSV</td><td>Lets you access all the motion status bits in one 32-bit word. This attribute is the same as the Motion Status tag.

<table><tr><th>Motion Status</th><th>Bit</th></tr><tr><td>Accel Status</td><td>0</td></tr><tr><td>Decel Status</td><td>1</td></tr><tr><td>Move Status</td><td>2</td></tr><tr><td>Jog Status</td><td>3</td></tr><tr><td>Gearing Status</td><td>4</td></tr><tr><td>Homing Status</td><td>5</td></tr><tr><td>Stopping Status</td><td>6</td></tr><tr><td>Homed Status</td><td>7</td></tr><tr><td>Position Cam Status</td><td>8</td></tr><tr><td>Time Cam Status</td><td>9</td></tr><tr><td>Position Cam Pending Status</td><td>10</td></tr><tr><td>Time Cam Pending Status</td><td>11</td></tr><tr><td>Gearing Lock Status</td><td>12</td></tr><tr><td>Position Cam Lock Status</td><td>13</td></tr><tr><td>Reserved</td><td>14</td></tr><tr><td>Master Offset Move Status</td><td>15</td></tr><tr><td>Coordinated Motion Status</td><td>16</td></tr></table></td></tr>
<tr><td>Motor Capacity</td><td>AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
Tag</td><td>Important: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1.

The present utilization of motor capacity as a percent of rated capacity.</td></tr>
<tr><td>Motor Data</td><td>AXIS_SERVO_DRIVE</td><td>Struct {
INT;
SINT
[256]}</td><td>MSG</td><td>Struct {length; data[ ]}

The Motor Data attribute is a structure with a length element and an array of bytes that contains important motor configuration information needed by an A-B SERCOS drive to operate the motor. The length element represents the number of valid data elements in the data array. The meaning of data within the data array is understood only by the drive. The block of data stored in the Motor Data attribute is derived at configuration time from an RSLogix 5000 motion database file.</td></tr>
<tr><td>Motor Electrical Angle</td><td>AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
Tag</td><td>Important: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1.

Degrees

The present electrical angle of the motor shaft.</td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Motor Feedback Configuration</td><td>AXIS_SERVO_DRIVE</td><td>INT</td><td>GSV</td><td>The controller and drive use this for scaling the feedback device counts. These attributes are derived from the corresponding Motor and Auxiliary Feedback Unit attributes.
Bit
0 = Feedback type
· 0 — rotary (default)
· 1 — linear
1 = (reserved)
2 = Linear feedback unit
· 0 — metric
· 1 — english
3 = Feedback Polarity (Aux Only)
· 0 — not inverted
· 1 — inverted

<table><tr><th colspan="3">If the bits are</th><th>Then Feedback Resolution is scaled to</th></tr><tr><th>2</th><th>1</th><th>0</th><th></th></tr><tr><td>0</td><td></td><td>0</td><td>Feedback Cycles per Feedback Rev</td></tr><tr><td>1</td><td></td><td>0</td><td>Feedback Cycles per Feedback Rev</td></tr><tr><td>0</td><td></td><td>1</td><td>Feedback Cycles per mm</td></tr><tr><td>1</td><td></td><td>1</td><td>Feedback Cycles per inch</td></tr></table>
Feedback Polarity
The Feedback Polarity bit attribute can be used to change the sense of direction of the feedback device. This bit is only valid for auxiliary feedback devices. When performing motor/feedback hookup diagnostics on an auxiliary feedback device using the MRHD and MAHD instructions, the Feedback Polarity bit is configured for the auxiliary feedback device to insure negative feedback into the servo loop. Motor feedback devices must be wired properly for negative feedback since the Feedback Polarity bit is forced to 0, or non-inverted.</td></tr>
<tr><td>Motor Feedback Interpolation Factor</td><td>AXIS_SERVO_DRIVE</td><td>DINT</td><td>GSV</td><td>Feedback Counts per Cycle

The Feedback Interpolation attributes establish how many Feedback Counts there are in one Feedback Cycle. The Feedback Interpolation Factor depends on both the feedback device and the drive feedback circuitry. Quadrature encoder feedback devices and the associated drive feedback interface typically support 4x interpolation, so the Interpolation Factor for these devices would be set to 4 Feedback Counts per Cycle (Cycles are sometimes called Lines). High Resolution Sin/Cosine feedback device types can have interpolation factors as high as 2048 Counts per Cycle. The product to the Feedback Resolution and the corresponding Feedback Interpolation Factor is the overall resolution of the feedback channel in Feedback Counts per Feedback Unit. In our example, a Quadrature encoder with a 2000 line/rev resolution and 4x interpolation factor would have an overall resolution of 8000 counts/rev.</td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Motor Feedback Resolution</td><td>AXIS_SERVO_DRIVE</td><td>DINT</td><td>GSV</td><td>Cycles per Motor Feedback Unit

The Motor and Aux Feedback Resolution attributes are used to provide the A-B drive with the resolution of the associated feedback device in cycles per feedback unit. These parameters provide the SERCOS drive with critical information needed to compute scaling factors used to convert Drive Counts to Feedback counts.</td></tr>
<tr><td>Motor Feedback Type</td><td>AXIS_SERVO_DRIVE</td><td>INT</td><td>GSV</td><td>The Motor and Aux Feedback Type attributes are used to identify the motor mounted or auxiliary feedback device connected to the drive.
<table>
<tr><th>Feedback Type</th><th>Code</th><th>Rotary Only</th><th>Linear Only</th><th>Rotary or Linear</th></tr>
<tr><td><None></td><td>0x0000</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>SRS</td><td>0x0001</td><td>X</td><td></td><td></td></tr>
<tr><td>SRM</td><td>0x0002</td><td>X</td><td></td><td></td></tr>
<tr><td>SCS</td><td>0x0003</td><td>X</td><td></td><td></td></tr>
<tr><td>SCM</td><td>0x0004</td><td>X</td><td></td><td></td></tr>
<tr><td>SNS</td><td>0x0005</td><td>X</td><td></td><td></td></tr>
<tr><td>MHG</td><td>0x0006</td><td>X</td><td></td><td></td></tr>
<tr><td>Resolver</td><td>0x0007</td><td>X</td><td></td><td></td></tr>
<tr><td>Analog Reference</td><td>0x0008</td><td>X</td><td></td><td></td></tr>
<tr><td>Sin/Cos</td><td>0x0009</td><td></td><td></td><td>X</td></tr>
<tr><td>TTL</td><td>0x000A</td><td></td><td></td><td>X</td></tr>
<tr><td>UVW</td><td>0x000B</td><td></td><td></td><td>X</td></tr>
<tr><td>Unknown Stegmann</td><td>0x000C</td><td></td><td></td><td>X</td></tr>
<tr><td>Endat</td><td>0x000D</td><td></td><td></td><td>X</td></tr>
<tr><td>RCM21S-4</td><td>0x000E</td><td>X</td><td></td><td></td></tr>
<tr><td>RCM21S-6</td><td>0x000F</td><td>X</td><td></td><td></td></tr>
<tr><td>RCM21S-8</td><td>0x0010</td><td>X</td><td></td><td></td></tr>
<tr><td>LINCODER</td><td>0x0011</td><td></td><td>X</td><td></td></tr>
<tr><td>Sin/Cos with Hall</td><td>0x0012</td><td></td><td></td><td>X</td></tr>
<tr><td>TTL with Hall</td><td>0x0013</td><td></td><td></td><td>X</td></tr>
</table></td></tr>
<tr><td>Motor Feedback Units</td><td>AXIS_SERVO_DRIVE</td><td>INT</td><td>GSV</td><td>The Motor Feedback Units attribute establishes the unit of measure that is applied to the Motor Feedback Resolution attribute value. The Aux Feedback Units attribute establishes the unit of measure that is applied to the Aux Feedback Resolution attribute value. Units appearing in the enumerated list cover linear or rotary, english or metric feedback devices.
0 = revs
1 = inches
2 = mm</td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Motor ID</td><td>AXIS_SERVO_DRIVE</td><td>INT</td><td>GSV</td><td>The Motor ID attribute contains the enumeration of the specific A-B motor catalog number associated with the axis. If the Motor ID does not match that of the actual motor, an error is generated during the drive configuration process.</td></tr>
<tr><td>Motor Inertia</td><td>AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
SSV</td><td>%Rated / Pos Units per Sec^2

The Motor Inertia value represents the inertia of the motor without any load attached to the motor shaft in Torque Scaling units of %Rated / Pos Units per Sec^2. The Load Inertia Ratio attribute's value represents the ratio of the load inertia to the motor inertia. Auto-tuning uses the Motor Inertia value to calculate the Load Inertia Ratio based on the following equation.

Load Inertia Ratio = (Total Inertia - Motor Inertia) / Motor Inertia.

Total Inertia is directly measured by the auto-tuning algorithm and applied to the Torque Scaling attribute in units of %Rated / Pos Units per Sec^2.

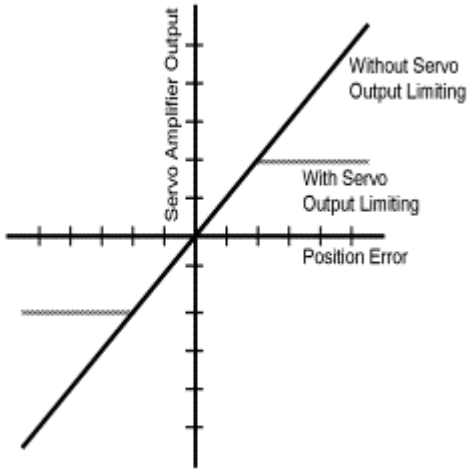
If the Load Inertia Ratio value is known, the Motor Inertia value can also be used to calculate a suitable Torque Scaling value for the fully loaded motor without performing an auto-tune. The equation used by RSLogix5000 to calculate the Torque Scaling value is as follows:

Torque Scaling = (1 + Load Inertia Ratio) * Motor Inertia.

The value for Load Inertia may be automatically calculated using Rockwell's MotionBook program while the value for Motor Inertia is derived from the Motion database file based on the motor selection.</td></tr>
<tr><td>Motor Overtemp Fault</td><td>AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>Set when the motor's temperature exceeds the motor shutdown temperature.</td></tr>
<tr><td>Motor Thermal Fault Action</td><td>AXIS_SERVO_DRIVE</td><td>SINT</td><td>GSV
SSV</td><td><table><tr><th>Fault Action</th><th>Value</th></tr><tr><td>Shutdown</td><td>0</td></tr><tr><td>Disable Drive</td><td>1</td></tr><tr><td>Stop Motion</td><td>2</td></tr><tr><td>Status Only</td><td>3</td></tr></table></td></tr>
<tr><td>Move Status</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>BOOL</td><td>Tag</td><td>Set if a Move motion profile is currently in progress. Cleared when the Move is complete or is superseded by some other motion operation.</td></tr>
<tr><td>Neg Dynamic Torque Limit</td><td>AXIS_SERVO_DRIVE</td><td>REAL</td><td>Tag</td><td>The currently operative negative positive torque/current limit magnitude. It should be the lowest value of all torque/current limits in the drive at a given time, including: amplifier peak limit, motor peak limit, user current limit, amplifier thermal limit, and motor thermal limit.</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Neg Hard Overtravel Fault	AXIS_SERVO_DRIVE	BOOL	Tag	Set if the axis moves beyond the negative direction position limits as established by hardware overtravel limit switches mounted on the equipment. This fault can only occur when the drive is in the enabled state and the Hard Overtravel Checking bit is set in the Fault Configuration Bits attribute. If the Hard Overtravel Fault Action is set for Stop Command, the faulted axis can be moved or jogged back inside the soft overtravel limits. Any attempt, however, to move the axis further beyond the hard overtravel limit switch using a motion instruction results in an instruction error. To recover from this fault, the axis must be moved back within normal operation limits of the equipment and the limit switch closed. This fault condition is latched and requires execution of an Motion Axis Fault Reset (MAFR) or Motion Axis Shutdown Reset (MASR) instruction to clear. Any attempt to clear the fault while the overtravel limit switch is still open and the drive is enabled is unsuccessful.
Neg Overtravel Input Status	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	If this bit is: · ON — The Negative Overtravel input is active. · OFF — The Negative Overtravel input is inactive.
Neg Soft Overtravel Fault	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	If this bit is: · ON — The axis moved or tried to move past the Maximum Negative travel limit. · OFF — The axis moved back within the Maximum Negative travel limit This fault can only happen when the drive is enabled and you configure the axis for Soft Travel Limits. If the Soft Overtravel Fault Action is set for Stop Command, the faulted axis can be moved or jogged back inside the soft overtravel limits. Any attempt, however, to move the axis further beyond the soft overtravel limit using a motion instruction results in an instruction error. As soon as the axis is moved back within the specified soft overtravel limits, the corresponding soft overtravel fault bit is automatically cleared. However the soft overtravel fault stays through any attempt to clear it while the axis position is still beyond the specified travel limits while the axis is enabled.
Negative Dynamic Torque Limit	AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important**: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. %Rated The currently operative maximum negative torque/current limit magnitude. The value should be the lowest value of all torque/current limits in the drive at a given time. This limit includes the amplifier peak limit, motor peak limit, user current limit, amplifier thermal limit, and the motor thermal limit.

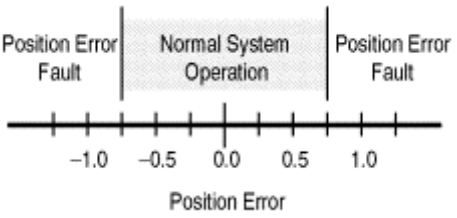
Attribute	Axis Type	Data Type	Access	Description
Output Cam Execution Targets	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	GSV	Represents the number of Output Cam nodes attached to this axis. Valid range = 0-8 with default of 0. The Output Cam Execution Targets attribute is used to specify the number of Output Cam nodes attached to the axis. This attribute can only be set as part of an axis create service and dictates how many Output Cam Nodes are created and associated to that axis. Each Output Cam Execution Target requires approximately 5.4k bytes of data table memory to store persistent data. With four Output Cam Execution Targets per axis, an additional 21.6k bytes of memory is required for each axis. The ability to configure the number of Output Cam Execution Targets for a specific axis reduces the memory required per axis for users who do not need Output Cam functionality, or only need 1 or 2 Output Cam Execution Targets for a specific axis. Each axis can be configured differently.
Output Cam Lock Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	GSV Tag	Set of Output Cam Lock Status bits The Output Cam Lock Status bit is set when an Output Cam has been armed. This would be initiated by executing an MAOC instruction with Immediate execution selected, when a pending output cam changes to armed, or when the axis approaches or passes through the specified axis arm position. As soon as this output cam current position moves beyond the cam start or cam stop position, the Output Cam Lock bit is cleared. This bit is also cleared if the Output Cam is terminated by a MDOC instruction.
Output Cam Lock Status	AXIS_CONSUMED AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	Tag	A set of bits that are set when an Output Cam is locked to the Master Axis. The bit number corresponds with the execution target number. One bit per execution target.
Output Cam Pending Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	GSV Tag	A set of bits that are set when an Output Cam is waiting for an armed Output Cam to move beyond its cam start/cam end position. The bit number corresponds with the execution target number. One bit per execution target. The Output Cam Pending Status bit is set if an Output Cam is currently pending the completion of another Output Cam. This would be initiated by executing an MAOC instruction with Pending execution selected. As soon as this output cam is armed, being triggered when the currently executing Output Cam has completed, the Output Cam Pending bit is cleared. This bit is also cleared if the Output Cam is terminated by a MDOC instruction.
Output Cam Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	GSV Tag	A set of bits that are set when the Output Cam has been initiated. The bit number corresponds with the execution target number. One bit per execution target. The Output Cam Status bit is set when an Output Cam has been initiated. The Output Cam Status bit is reset when the cam position moves beyond the cam start or cam end position in “Once” execution mode with no Output Cam pending or when the Output Cam is terminated by a MDOC instruction.

Attribute	Axis Type	Data Type	Access	Description
Output Cam Transition Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	GSV Tag	A set of bits that are set when the transition from the current armed Output Cam to the pending Output Cam is in process. The bit number corresponds with the execution target number. One bit per execution target. The Output Cam Transition Status bit is set when a transition between the currently armed and the pending Output Cam is in process. Therefore, each Output Cam controls a subset of Output Bits. The Output Cam Transition Status bit is reset, when the transition to the pending Output Cam is complete or when the Output Cam is terminated by a MDOC instruction.
Output Limit	AXIS_SERVO	REAL	GSV SSV	0.0…10.0V The Output Limit attribute provides a method of limiting the maximum servo output voltage of a physical axis to a specified level. The servo output for the axis as a function of position servo error, both with and without servo output limiting, is shown below.  The servo output limit may be used as a software current or torque limit if you are using a servo drive in torque (current) loop mode. The percentage of the drive's maximum current that the servo controller commands is equal to the specified servo output limit. For example, if the drive is capable of 30 Amps of current for a 10 Volt input, setting the servo output limit to 5V limits the maximum drive current to 15 Amps. The servo output limit may also be used if the drive cannot accept the full ±10 Volt range of the servo output. In this case, the servo output limit value effectively limits the maximum command sent to the amplifier. For example, if the drive can only accept command signals up to ±7.5 Volts, set the servo output limit value to 7.5 volts.
Output Limit Status	AXIS_SERVO	BOOL	Tag	If this bit is: · ON — The servo output is at or past the Output Limit value. · OFF — The servo output is within the Output Limit value

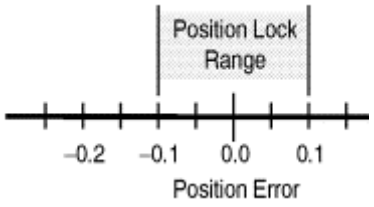
Attribute	Axis Type	Data Type	Access	Description
Output LP Filter Bandwidth	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	Hertz The Output LP (Low Pass) Filter Bandwidth controls the bandwidth of the drive's low-pass digital output filter. The programmable low-pass output filter is bypassed if the configured Output LP Filter Bandwidth for this filter is set to zero (the default). This output filter can be used to filter out, or reduce, high frequency variation of the drive output to the motor. The lower the Output LP Filter Bandwidth, the greater the attenuation of these high frequency components of the output signal. Unfortunately, since the low-pass filter adds lag to the servo loop which pushes the system towards instability, decreasing the Output LP Filter Bandwidth usually requires lowering the Position or Velocity Proportional Gain of the system to maintain stability. The output filter is particularly useful in high inertia applications where resonance behavior can severely restrict the maximum bandwidth capability of the servo loop.
Output Notch Filter Frequency	AXIS_SERVO_DRIVE	REAL	GSV SSV	Hertz The Output Notch Filter Frequency attribute controls the center frequency of the drive's digital notch filter. Currently implemented as a 2^{nd} order digital filter with a fixed Q, the Notch Filter provides approximately 40DB of output attenuation at the Notch Filter Frequency. The programmable notch filter is bypassed if the configured Output Notch Filter Frequency for this filter is set to zero (the default). This output notch filter is particularly useful in attenuating mechanical resonance phenomena. The output filter is particularly useful in high inertia applications where mechanical resonance behavior can severely restrict the maximum bandwidth capability of the servo loop.
Output Offset	AXIS_SERVO	REAL	GSV SSV	+/-10V Another common situation when interfacing an external Servo Drive, particularly for velocity servo drives, is the effect of drive offset. Cumulative offsets of the servo module's DAC output and the Servo Drive Input result in a situation where a zero commanded Servo Output value causes the axis to "drift". If the drift is excessive it can play havoc on the Hookup Diagnostic and Tuning procedures as well as result in a steady-state non-zero position error when the servo loop is closed. Output offset compensation can be used to correct this problem by adding a fixed value, called Output Offset, to the Servo Output. This value is chosen to achieve near zero drive velocity when the uncompensated Servo Output value is zero.
Overload Fault	AXIS_SERVO_DRIVE	BOOL	Tag	When the load limit of the motor/drive is first exceeded, the Overload warning bit is set. If the condition persists, the Overload fault is set. Often this bit is tied into the IT limit of the drive.
Overspeed Fault	AXIS_SERVO_DRIVE	BOOL	Tag	Set when the speed of the axis as determined from the feedback has exceeded the overspeed limit which is typically set to 150% of configured velocity limit for the motor.

Attribute	Axis Type	Data Type	Access	Description
Physical Axis Fault	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	If this bit is set, the physical axis has one or more faults. The specific faults can then be determined through access to the fault attributes of the associated physical axis. Do you want this fault to give the controller a major fault? · YES — Set the General Fault Type of the motion group = Major Fault. · NO — You must write code to handle these faults.
Pos Dynamic Torque Limit	AXIS_SERVO_DRIVE	REAL	Tag	The currently operative maximum positive torque/current limit magnitude. It should be the lowest value of all torque/current limits in the drive at a given time, including: amplifier peak limit, motor peak limit, user current limit, amplifier thermal limit, and motor thermal limit.
Pos Hard Overtravel Fault	AXIS_SERVO_DRIVE	BOOL	Tag	Set if the axis moves beyond the current position limits as established by hardware overtravel limit switches mounted on the equipment. This fault can only occur when the drive is in the enabled state and the Hard Overtravel Checking bit is set in the Fault Configuration Bits attribute. If the Hard Overtravel Fault Action is set for Stop Command, the faulted axis can be moved or jogged back inside the soft overtravel limits. Any attempt, however, to move the axis further beyond the hard overtravel limit switch using a motion instruction results in an instruction error. To recover from this fault, the axis must be moved back within normal operation limits of the equipment and the limit switch closed. This fault condition is latched and requires execution of an Motion Axis Fault Reset (MAFR) or Motion Axis Shutdown Reset (MASR) instruction to clear. Any attempt to clear the fault while the overtravel limit switch is still open and the drive is enabled is unsuccessful.
Pos Lock Status	AXIS_SERVO AXIS_SERVO_DRIVE	DINT	Tag	Set when the magnitude of the axis position error has become less than or equal to the configured Position Lock Tolerance value for the associated physical axis.
Pos Overtravel Input Status	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	If this bit is: · ON — The Positive Overtravel input is active. · OFF — The Positive Overtravel input is inactive.
Pos Soft Overtravel Fault	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	If this bit is: · ON — The axis moved or tried to move past the Maximum Positive travel limit. · OFF — The axis moved back within the Maximum Positive travel limit This fault can only happen when the drive is enabled and you configure the axis for Soft Travel Limits. If the Soft Overtravel Fault Action is set for Stop Command, the faulted axis can be moved or jogged back inside the soft overtravel limits. Any attempt, however, to move the axis further beyond the soft overtravel limit using a motion instruction results in an instruction error. As soon as the axis is moved back within the specified soft overtravel limits, the corresponding soft overtravel fault bit is automatically cleared. However the soft overtravel fault stays through any attempt to clear it while the axis position is still beyond the specified travel limits while the axis is enabled.

Attribute	Axis Type	Data Type	Access	Description
Position Cam Lock Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set whenever the master axis satisfies the starting condition of a currently active Position Cam motion profile. The starting condition is established by the Start Control and Start Position parameters of the MAPC instruction. This bit is bit is cleared when the current position cam profile completes, or is superseded by some other motion operation. In unidirectional master direction mode, the Position Cam Lock Status bit is cleared when moving in the "wrong" direction and sets when moving in the "correct" direction.
Position Cam Pending Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set if a Position Cam motion profile is currently pending the completion of a currently executing cam profile. This would be initiated by executing an MAPC instruction with Pending execution selected. This bit is cleared when the current position cam profile completes, initiating the start of the pending cam profile. This bit is also cleared if the position cam profile completes, or is superseded by some other motion operation.
Position Cam Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set if a Position Cam motion profile is currently in progress. Cleared when the Position Cam is complete or is superseded by some other motion operation.
Position Command	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV Tag	Position Command in Position Units **Important**: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. Position Command is the current value of the Fine Command Position into the position loop summing junction, in configured axis Position Units. Within the active servo loop, the Position Command value is used to control the position of the axis.
Position Data Scaling	AXIS_SERVO_DRIVE	INT	GSV	This attribute is derived from the Drive Units attribute. See IDN 76 in IEC 1491.
Position Data Scaling Exp	AXIS_SERVO_DRIVE	INT	GSV	This attribute is derived from the Drive Units attribute. See IDN 78 in IEC 1491.
Position Data Scaling Factor	AXIS_SERVO_DRIVE	DINT	GSV	This attribute is derived from the Drive Units attribute. See IDN 77 in IEC 1491.
Position Differential Gain	AXIS_SERVO	REAL	GSV SSV	In some External Velocity Servo Drive applications where the level of damping provided by the external drive is insufficient for good position servo loop performance, additional damping may be achieved via the Position Loop Differential Gain. Assuming a non-zero Position Loop Differential Gain value, the difference between the current Position Error value and the last Position Error value is computed. This value is then multiplied by the Position Loop Differential Gain to produce a component to the Servo Output or Velocity Command that attempts to correct for the change in position error, creating a "damping" effect. Increasing this gain value results in greater "damping" of the axis.

Attribute	Axis Type	Data Type	Access	Description
Position Error	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important**: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. Position Error in Position Units Position Error is the difference, in configured axis Position Units, between the command and actual positions of an axis. For an axis with an active servo loop, position error is used, along with other error terms, to drive the motor to the condition where the actual position is equal to the command position.
Position Error Fault	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	Set when the axis position error exceeds the Position Error Tolerance. This fault can only occur when the drive is in the enabled state. The controller latches this fault. Use a Motion Axis Fault Reset (MAFR) or Motion Axis Shutdown Reset (MASR) instruction to clear the fault.
Position Error Fault Action	AXIS_SERVO AXIS_SERVO_DRIVE	SINT	GSV SSV	**Fault Action** / **Value** Shutdown / 0 Disable Drive / 1 Stop Motion / 2 Status Only / 3
Position Error Tolerance	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units The Position Error Tolerance parameter specifies how much position error the servo or drive tolerates before issuing a Position Error Fault. Like the position lock tolerance, the position error tolerance is interpreted as a ± quantity. For example, specifying a position error tolerance of 0.75 Position Units means that a Position Error Fault is generated whenever the position error of the axis is greater than 0.75 or less than -0.75 Position Units, as shown.  The self tuning routine sets the position error tolerance to twice the following error at maximum speed based on the measured response of the axis. In most applications, this value provides reasonable protection in case of an axis fault or stall condition without nuisance faults during normal operation. If you need to change the calculated position error tolerance value, the recommended setting is 150% to 200% of the position error while the axis is running at its maximum speed.

Attribute	Axis Type	Data Type	Access	Description
Position Feedback	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important**: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. Position Feedback in Position Units Position Feedback is the current value of the Fine Actual Position into the position loop summing junction, in configured axis Position Units. Within the servo loop, the Position Feedback represents the current position of the axis.
Position Integral Gain	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	1/mSec-Sec Position Integral Gain (Pos I Gain) improves the steady-state positioning performance of the system. By using Position Integral Gain, it is possible to achieve accurate axis positioning despite the presence of such disturbances as static friction or gravity. Increasing the integral gain generally increases the ultimate positioning accuracy of the system. Excessive integral gain, however, results in system instability. Every servo update, the current Position Error is accumulated in a variable called the Position Integral Error. This value is multiplied by the Position Integral Gain to produce a component to the Velocity Command that attempts to correct for the position error. The characteristic of Pos I Gain correction, however, is that any non-zero Position Error accumulates in time to generate enough force to make the correction. This attribute of Pos I Gain makes it invaluable in applications where positioning accuracy or tracking accuracy is critical. The higher the Pos I Gain value the faster the axis is driven to the zero Position Error condition. Unfortunately, Pos I Gain control is intrinsically unstable. Too much Pos I Gain results in axis oscillation and servo instability. If the axis is configured for an external velocity loop servo drive, the Pos I Gain should be zero–most analog velocity loop servo amplifiers have integral gain of their own and do not tolerate *any* amount of Pos I Gain in the position loop without producing severe oscillations. If Pos I Gain is necessary for the application, the velocity integrator in the drive must be disabled. In certain cases, Pos I Gain control is disabled. One such case is when the servo output to the axis' drive is saturated. Continuing integral control behavior in this case would only exacerbate the situation. Another common case is when performing certain motion. When the Integrator Hold Enable attribute is set, the servo loop automatically disables the integrator during commanded motion. While the Pos I Gain, if employed, is typically established by the automatic servo tuning procedure, the Pos I Gain value may also be set manually. You can compute the Pos I Gain based on the current or computed value for the Pos P Gain using the following formula: $\text{Pos I Gain} = 0.25 * 0.001 \text{ Sec/mSec} * (\text{Pos P Gain})^2$ Assuming a Pos P Gain value of 100 Sec^{-1} this results in a Pos I Gain value of 2.5 ~0.1 mSec^{-1}-Sec^{-1}

Attribute	Axis Type	Data Type	Access	Description
Position Integrator Error	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important**: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. Position Integrator Error in Position Units - mSec Position Integrator Error is the running sum of the Position Error, in the configured axis Position Units, for the specified axis. For an axis with an active servo loop, the position integrator error is used, along with other error terms, to drive the motor to the condition where the actual position is equal to the command position.
Position Lock Status	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	If this bit is: · ON — The axis position error is less than or equal to the Position Lock Tolerance value of the axis. · OFF — The axis position error is greater than the Position Lock Tolerance value of the axis.
Position Lock Tolerance	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units The Position Lock Tolerance attribute value specifies how much position error the motion module tolerates when giving a true Position Locked Status indication. When used in conjunction with the Position Locked Status bit, it is a useful parameter to control positioning accuracy. The Position Lock Tolerance value should be set, in Position Units, to the desired positioning accuracy of the axis. Note that the position lock tolerance value is interpreted as a ± quantity. For example, if your position units are Inches, specifying a position lock tolerance of 0.01 provides a minimum positioning accuracy of ±0.01 inches as shown below. 
Position Polarity	AXIS_SERVO_DRIVE	INT	GSV	This attribute is derived from the Drive Polarity attribute. See IDN 55 in IEC 1491.

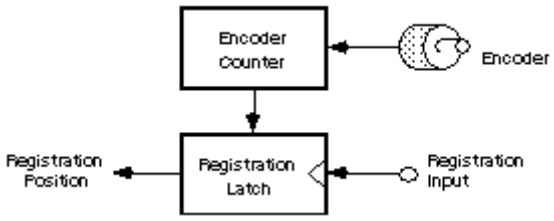
Attribute	Axis Type	Data Type	Access	Description
Position Proportional Gain	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	1/Sec The Position Error is multiplied by the Position Proportional Gain (Pos P Gain) to produce a component to the Velocity Command that tries to correct for the position error. Increasing this gain increases the bandwidth of the position servo loop and results in greater static stiffness of the axis, which is a measure of the corrective force that is applied to an axis for a given position error. Too little Pos P Gain results in excessively compliant, or mushy, axis behavior. Too large a Pos P Gain results in axis oscillation due to servo instability. A well-tuned system moves and stops quickly and shows little or no ringing during constant velocity or when the axis stops. If the response time is poor, or the motion sloppy or slow, you may need to increase the proportional gain. If excessive ringing or overshoot is observed when the motor stops, you may need to decrease the proportional gain. While the tuning procedure sets the Pos P Gain, you can also set it manually. You can compute the Pos P Gain based on either the desired loop gain or the desired bandwidth of the position servo system. **Loop Gain Method** If you know the desired loop gain in Inches per Minute per mil or millimeters per minute per mil, use the following formula to calculate the corresponding P gain. Pos P Gain = 16.667 * Desired Loop Gain (IPM/mil) A loop gain of 1 IPM/mil (Pos P gain = 16.7 Sec^{-1}) gives stable positioning for most axes. However, position servo systems typically run much tighter than this. The typical value for the Position Proportional Gain is ~100 Sec^{-1}. **Bandwidth Method** If you know the desired unity gain bandwidth of the position servo in Hertz, use the following formula to calculate the corresponding P gain. Pos P Gain = Bandwidth (Hertz) / 6.28 Position servo systems typically run with at least a unity gain bandwidth of ~16 Hertz. The typical value for the Position Proportional Gain is ~100 Sec^{-1}. **Maximum Bandwidth** There are limitations to the maximum bandwidth that can be achieved for the position loop based on the dynamics of the inner velocity and torque loops of the system and the desired damping of the system, Z. These limitations may be expressed as follows: Bandwidth (Pos) = 0.25 * $1/Z^2$ * Bandwidth (Vel) = 0.25 * $1/Z^2$ * Bandwidth (Torque) For example, if the bandwidth of the drive's torque loop is 100 Hz and the damping factor, Z, is 0.8, the velocity bandwidth is approximately 40 Hz and the position bandwidth is 16 Hz. Based on these numbers the corresponding proportional gains for the loops can be computed. Note that the bandwidth of the torque loop includes feedback sampling delay and filter time constant.

Attribute	Axis Type	Data Type	Access	Description
Position Servo Bandwidth	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	Hertz The value for the Position Servo Bandwidth represents the unity gain bandwidth that is to be used to calculate the gains for a subsequent MAAT (Motion Apply Axis Tune) instruction. The unity gain bandwidth is the frequency beyond which the position servo is unable to provide any significant position disturbance correction. In general, within the constraints of a stable servo system, the higher the Position Servo Bandwidth is the better the dynamic performance of the system. A maximum value for the Position Servo Bandwidth is generated by the MRAT (Motion Run Axis Tune) instruction. Computing gains based on this maximum value via the MAAT instruction results in dynamic response in keeping with the current value of the Damping Factor described above. Alternatively, the responsiveness of the system can be "softened" by reducing the value of the Position Servo Bandwidth before executing the MAAT instruction.. There are limitations to the maximum bandwidth that can be achieved for the position loop based on the dynamics of the inner velocity and current loops of the servo system and the desired damping of the system, Z. Exceeding these limits could result in an unstable system. These bandwidth limitations may be expressed as follows: Max Position Bandwidth (Hz) = 0.25 * $1/Z^2$ * Velocity Bandwidth (Hz) For example, if the maximum bandwidth of the velocity servo loop is 40 Hz and the damping factor, Z, is 0.8, the maximum the maximum position bandwidth is 16 Hz. Based on these numbers the corresponding proportional gains for the loops can be computed.
Position Units	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	STRING	MSG	Fixed length string of 32 characters The Position Units attribute can support an ASCII text string of up to 32 characters. This string is used by RSLogix 5000 software in the axis configuration dialogs to request values for motion-related parameters in the specified Position Units.
Position Unwind	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	GSV SSV	Counts per Revolution If the axis is configured as a rotary axis by setting the corresponding Rotary Axis bit Servo Configuration Bit word, a value for the Position Unwind attribute is required. This is the value used to perform automatic electronic unwind of the rotary axis. Electronic unwind allows infinite position range for rotary axes by subtracting the unwind value from both the actual and command position every time the axis makes a complete revolution. To avoid accumulated error due to round-off with irrational conversion constants the unwind value is requested in units feedback counts per axis revolution and must be an integer. For example, suppose that a given axis is configured as a Rotary Axis with Position Units of "Degrees" and 10 feedback counts per degree. It is desired to unwind the axis position after every revolution. In this case, the Position Unwind attribute should be set to 3600 since there are 3600 feedback counts (10 * 360) per revolution of the axis.

Attribute	Axis Type	Data Type	Access	Description
Positive Dynamic Torque Limit	AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important**: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. %Rated The currently operative maximum positive torque/current limit magnitude. The value should be the lowest value of all torque/current limits in the drive at a given time. This limit includes the amplifier peak limit, motor peak limit, user current limit, amplifier thermal limit, and the motor thermal limit.
Power Capacity	AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important**: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. The present utilization of the axis power supply as a percent of rated capacity.
Power Limit Status	AXIS_SERVO_DRIVE	BOOL	Tag	Set when the magnitude of the actual supplied power is greater than the configured Power Threshold.
Power Phase Loss Fault	AXIS_SERVO_DRIVE	BOOL	Tag	Set when the drive detects that one or more of the three power line phases is lost from the 3 phase power inputs.
Power Supply ID	AXIS_SERVO_DRIVE	INT	GSV	The Power Supply ID attribute contains the enumeration of the specific A-B Power Supply or System Module catalog numbers associated with the axis. If the Power Supply ID does not match that of the actual supply hardware, an error is generated during the drive configuration process.
Precharge Overload Fault	AXIS_SERVO_DRIVE	BOOL	Tag	The drive's precharge resistor gets too hot if you cycle 3-phase power too many times. If that happens, this bit turns on.
Primary Operation Mode	AXIS_SERVO_DRIVE	INT	GSV	This attribute is derived from the Servo Loop Configuration attribute. See IDN 32 in IEC 1491.
Process Status	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	Set when there is an axis tuning operation or an axis hookup diagnostic test operation in progress on the axis.

Attribute	Axis Type	Data Type	Access	Description
Programmed Stop Mode	AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	SINT	GSV SSV	Determines how a specific axis will stop when the controller has a critical controller mode change or when an MGS (Motion Group Stop) instruction executes with it's stop mode set to Programmed. The modes fo the controller are: Program Mode, Run Mode, Test Mode, and Faulted Mode. Any mode change into or out of program mode (prog->run, prog->test, run->prog & test->prog) will initiate a programmed stop for every axis owned by that controller. Each individual axis can have its own Programmed Stop Mode configuration independent of other axes. **Fast Stop (default) = 0** When the Programmed Stop Mode attribute is configured for Fast Stop, the axis is decelerated to a stop using the current configured value for Maximum Deceleration. Servo action is maintained after the axis motion has stopped. **Fast Disable = 1** When the Programmed Stop Mode attribute is configured for Fast Disable, the axis is decelerated to a stop using the current configured value for Maximum Deceleration. Servo action is maintained until the axis motion has stopped at which time the axis is disabled, that is, Drive Enable disabled, and Servo Action disabled **Hard Disable = 2** When configured for Hard Disable, the axis is immediately disabled, that is, Drive Enable disabled, Servo Action disabled, but the OK contact is left closed. Unless the drive is configured to provide some form of dynamic breaking, this results in the axis coasting to a stop. **Fast Shutdown = 3** When configured for Fast Shutdown, the axis is decelerated to a stop as with Fast Stop but, once the axis motion is stopped, the axis is placed in the Shutdown state, that is, Drive Enable disabled, servo action disabled, and the OK contact opened. To recover from the Shutdown state requires execution of one of the axis or group Shutdown Reset instructions (MASR or MGSR). **Hard Shutdown = 4** When configured for Hard Shutdown, the axis is immediately placed in the Shutdown state, that is, Drive Enable disabled, Servo Action disabled, and the OK contact opened. Unless the drive is configured to provide some form of dynamic breaking, this results in the axis coasting to a stop. To recover from the Shutdown state requires execution of one of the axis or group Shutdown Reset instructions (MASR or MGSR).
PWM Frequency Select	AXIS_SERVO_DRIVE	SINT	GSV	The PWM Frequency Select attribute controls the frequency of the pulse width modulated voltage applied to the motor by the drive's power structure. Higher PWM Frequency values reduce torque ripple and motor noise based on the motor's electrical time constant. Higher PWM frequencies, however, mean higher switching frequencies, which tends to produce more heat in the drive's power structure. So, for applications that have high torque demands, a lower PWM frequency would be more appropriate. 0 = low frequency (default) 1 = high frequency

Attribute	Axis Type	Data Type	Access	Description
Reg 1 Input Status	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	If this bit is: · ON — Registration 1 input is active. · OFF — Registration 1 input is inactive.
Reg 2 Input Status	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	If this bit is: · ON — Registration 2 input is active. · OFF — Registration 2 input is inactive.
Reg Event 1 Armed Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set when a registration checking has been armed for registration input 1 through execution of the MAR (Motion Arm Registration) instruction. Cleared when either a registration event occurs or a MDR (Motion Disarm Registration) instruction is executed for registration input 1.
Reg Event 1 Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set when a registration event has occurred on registration input 1. Cleared when either another MAR (Motion Arm Registration) instruction or a MDR (Motion Disarm Registration) instruction is executed for registration input 1.
Reg Event 2 Armed Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set when a registration checking has been armed for registration input 2 through execution of the MAR (Motion Arm Registration) instruction. Cleared when either a registration event occurs or a MDR (Motion Disarm Registration) instruction is executed for registration input 2.
Reg Event 2 Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set when a registration event has occurred on registration input 2. Cleared when either another MAR (Motion Arm Registration) instruction or a MDR (Motion Disarm Registration) instruction is executed for registration input 2.
Registration 1 Position	AXIS_CONSUMED AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	Tag	Registration 1 Position in Position Units
Registration 1 Event Task Registration 2 Event Task	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	MSG	These attributes show which task is triggered when the registration event happens. · An instance of 0 means that no event task is configured to be triggered by the registration event. · The task is triggered at the same time that the Process Complete bit is set for the instruction that armed the watch event. · The controller sets these attributes. Don't set them by an external device.

Attribute	Axis Type	Data Type	Access	Description
Registration 1 Position Registration 2 Position	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	Position Units Two registration position attributes are provided to independently store axis position associated with two different registration input events. The Registration Position value is the absolute position of a physical or virtual axis (in the position units of that axis) at the occurrence of the most recent registration event for that axis. The figure below shows how the registration position is latched by the registration input when a registration event occurs. The latching mechanism can be implemented in the controller software (soft registration) or, for greater accuracy, in physical hardware (hard registration).  The Registration Latch mechanism is controlled by two Event Control instructions, MAR (Motion Arm Registration) and MDR (Motion Disarm Registration). The accuracy of the registration position value, saved as a result of a registration event, is a function of the delay in recognizing the specified transition (typically 1 µsec for hardware registration) and the speed of the axis during this time. The uncertainty in the registration position is the distance traveled by the axis during this interval as shown by the equation. $$\text{Uncertainty} = \text{Axis Speed}\left[\frac{\text{Position Units}}{\text{Second}}\right] \times \text{Delay}$$ Use the formula given above to calculate the maximum registration position error for the expected axis speed. Alternatively, you can calculate the maximum axis speed for a specified registration accuracy by re-arranging this formula as shown. $$\text{Maximum Speed}\left[\frac{\text{Position Units}}{\text{Second}}\right] = \frac{\text{Desired Accuracy [Position Units]}}{\text{Delay}}$$
Registration 1 Time Registration 2 Time	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	GSV Tag	Lower 32 bits of CST time The two Registration Time values contain the lower 32-bits of CST time at which their respective registration events occurred. Units for this attribute are in microseconds.

Attribute	Axis Type	Data Type	Access	Description
Resistive Brake Contact Delay	AXIS_SERVO_DRIVE	REAL	GSV SSV	Sec This attribute controls an optional external Resistive Brake Module (RBM). The RBM is between the drive and the motor and uses an internal contactor to switch the motor between the drive and a resistive load. The drive's RBM output controls this contactor. When the drive's RBM output is energized, the RBM contactor is switched from the load resistors to the UVW motor lines connecting the drive to the motor. This switching does not occur instantaneously and enabling the power structure too early can cause electrical arcing across the contactor. The resistive brake contact delay is the time that it takes to fully close the contactor across the UVW motor lines. In order to prevent electrical arcing across the the contactor the enabling of the drive's power structure is delayed. The delay time is variable depending on the RBM model. When applying an RBM, you must set the Resistive Brake Contact Delay to the recommended value found in the RBM specification. The following cases outline how the RBM output relates to the normal enable and disable sequences. **Case 1 – Enable Sequence:** 1. Enable axis is initiated via MSO or MAH instruction. 2. Turn on RBM output to connect motor to drive. 3. Wait for Resistive Brake Contact Delay while RBM contacts close. 4. Drive power structure enabled (Drive Enable Status bit is set). 5. Turn on motor brake output to release brake. 6. Wait Brake Release Delay Time while motor brake releases. 7. Track Command reference (Servo Action Status bit is set). **Case 2 – Disable - Category 1 Stop** 1. Disable axis is initiated via an MSF instruction or a drive disable fault action. 2. Drive stops tracking command reference (Servo Action Status bit is cleared). 3. Apply Stopping Torque to stop motor. 4. Wait for zero speed or Stopping Time Limit. 5. Turn off brake output to engage motor brake. 6. Wait for Brake Engage delay while motor brake engages. 7. Disable drive power structure (Drive Enable Status bit is cleared). 8. Turn off RBM output to disconnect motor from drive. **Case 3 – Shutdown Category 0 Stop** 1. Drive stops tracking command reference (Servo Action Status bit is cleared). 2. Disable drive power structure (Drive Enable Status bit is cleared). 3. Turn off brake output to engage brake. 4. Turn off RBM output to disconnect motor from drive.

Attribute	Axis Type	Data Type	Access	Description
Rotary Axis	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	SINT	GSV SSV	0 = Linear 1 = Rotary When the Rotary Axis attribute is set true (1), it lets the axis unwind. This gives infinite position range by unwinding the axis position whenever the axis moves through a complete physical revolution. The number of encoder counts per physical revolution of the axis is specified by the Position Unwind attribute. For Linear operation, the counts don't roll over. They are limited to +/- 2 billion.
Safe-Off Mode Active Status	AXIS_SERVO_DRIVE	BOOL	GSV Tag	This bit is the status indication of the Kinetix Drive's Safe-Off circuitry. If this bit is the following state: • ON - The Drive's Safety monitor cicuitry has encountered a loss of signal from Enable_1 or Enable_2. • OFF - The Drive's Safety monitor circuitry has no fault from Enable_1 or Enable_2. For the Kinetix Drive to pass back this status to the controller via this bit, the Drive must have fimrware version 1.85 or higher.
SERCOS Error Code	AXIS_SERVO_DRIVE	INT	GSV* Tag	Error code returned by SERCOS module indicating source of drive parameter update failure. The SERCOS Error Code value can be used to identify the source of the drive parameter update failure that resulted in the Axis Configuration Fault. The error codes for this attribute are derived from the IEC-1394 SERCOS Interface standard.
SERCOS Fault	AXIS_SERVO_DRIVE	BOOL	Tag	Set when either a requested SERCOS procedure fails to execute properly or the associated drive node has detected a SERCOS communication fault.
SERCOS Ring Fault	AXIS_SERVO_DRIVE	BOOL	Tag	If this bit is set, there is a problem on the SERCOS ring; that is, the light has been broken or a drive has been powered down.
Servo Action Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	If this bit is: · ON — The axis is under servo control. · OFF — Servo action is disabled.

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Servo Fault</td><td>AXIS_SERVO</td><td>DINT</td><td>Tag</td><td>Lets you access all the servo fault bits in one 32-bit word. This tag is the same as the Servo Fault Bits attribute.

<table><tr><th>Servo Fault</th><th>Bit</th></tr><tr><td>Pos Soft Overtravel Fault</td><td>0</td></tr><tr><td>Neg Soft Overtravel Fault</td><td>1</td></tr><tr><td>Reserved</td><td>2</td></tr><tr><td>Reserved</td><td>3</td></tr><tr><td>Feedback Fault</td><td>4</td></tr><tr><td>Feedback Noise Fault</td><td>5</td></tr><tr><td>Reserved</td><td>6</td></tr><tr><td>Reserved</td><td>7</td></tr><tr><td>Position Error Fault</td><td>8</td></tr><tr><td>Drive Fault</td><td>9</td></tr></table>
These fault bits are updated every coarse update period.
Do you want any of these faults to give the controller a major fault?
· YES — Set the General Fault Type of the motion group = Major Fault.
· NO — You must write code to handle these faults.</td></tr>
<tr><td>Servo Fault Bits</td><td>AXIS_SERVO</td><td>DINT</td><td>GSV*</td><td>Lets you access all the servo fault bits in one 32-bit word. This attribute is the same as the Servo Fault tag.

<table><tr><th>Servo Fault</th><th>Bit</th></tr><tr><td>Pos Soft Overtravel Fault</td><td>0</td></tr><tr><td>Neg Soft Overtravel Fault</td><td>1</td></tr><tr><td>Reserved</td><td>2</td></tr><tr><td>Reserved</td><td>3</td></tr><tr><td>Feedback Fault</td><td>4</td></tr><tr><td>Feedback Noise Fault</td><td>5</td></tr><tr><td>Reserved</td><td>6</td></tr><tr><td>Reserved</td><td>7</td></tr><tr><td>Position Error Fault</td><td>8</td></tr><tr><td>Drive Fault</td><td>9</td></tr></table>
These fault bits are updated every coarse update period.
Do you want any of these faults to give the controller a major fault?
· YES — Set the General Fault Type of the motion group = Major Fault.
· NO — You must write code to handle these faults.</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Servo Feedback Type	AXIS_SERVO	SINT	GSV	This attribute provides a selection for the Feedback Type. 0 = A Quadrature B (AQB) 1 = Synchronous Serial Interface (SSI) 2 = Linear Displacement Transducer (LDT) **A Quadrature B Encoder Interface (AQB)** Servo modules, such as the 175-6M02AE, provide interface hardware to support incremental quadrature encoders equipped with standard 5-Volt differential encoder interface signals. This interface hardware provides a robust differential encoder input interface to condition each of the encoder signals before being applied to an Encoder-to-Digital Converter (EDC) FPGA. The EDC decodes the encoder signals and uses a 16-bit bidirectional counter to accumulate feedback counts. A regular Timer Event signal, applied to the EDC, latches the encoder counters for all axes simultaneously. This same Timer Event signal also triggers the servo interrupt service routine that performs the servo loop computations. One of the first things done by the interrupt service routine is to read the latched encoder counter values from the EDC. The change in the encoder counter value from the last timer event is computed and this delta value is added to a 32-bit signed integer position accumulator, which represents the Actual Position of the axis. The Actual Position value is used as feedback to the position servo loop and as input to the Watch Event Handler. The delta position value represents velocity feedback, which when configured to do so, may be filtered and applied to the inner velocity servo loop. **Synchronous Serial Interface (SSI)** Some servo modules, like the 1756-M02AS, provide an interface to transducers with Synchronous Serial Interface (SSI) outputs. SSI outputs use standard 5V differential signals (RS422) to transmit information from the transducer to the controller. The signals consist of a Clock generated by the controller and Data generated by the transducer. Each transducer with an SSI output provides output data of a specified number of bits of either Binary or Gray code data. The controller must generate a stream of clock pulses with the correct number of bits and a frequency within the range supported by the transducer. The servo module can be configured via the Servo Axis Object to generate any number of clock pulses between 8 and 32, and the frequency can be set to either 208kHz or 650kHz. The clock signal is maintained in the High state between pulse strings. The transducer shifts data out on the Data line MSB first on each rising edge of the clock signal. The transducer also maintains the data signal in specified states before and after the data is shifted out. These states are checked by the controller to detect missing transducers or broken wires. A Field Programmable Gate Array (FPGA) is used to implement a multi-channel SSI Interface on the controller. Each channel is functionally equivalent. ***Continued on next page***

Attribute	Axis Type	Data Type	Access	Description
Servo Feedback Type (cont.)				**Linear Displacement Transducer (LDT)** Servo modules like the 1756-HYD02 use the Linear Magnetostrictive Displacement Transducer, or LDT. A Field Programmable Gate Array (FPGA) is used to implement a multi-channel LDT Interface. Each channel is functionally equivalent and is capable of interfacing to an LDT device with a maximum count of 240,000. The LDT interface has transducer failure detection and digital filtering to reduce electrical noise. The FPGA can interface to two types of LDTs: Start/Stop and PWM. Start/Stop transducers accept an input (interrogate) signal to start the measurement cycle and respond with two pulses on the Return line. The time between the pulses is proportional to the position. PWM transducers respond to the interrogate signal with a single long pulse on the Return line. The pulse width is proportional to the position. The FPGA generates the Interrogate signal every Servo Update time and measures the time between the Start/Stop pulses or the PWM pulse width. The resolution of the position measurement is determined by the frequency of the clock used for the time measurement. In the 1756-HYD02 design, a 60 MHz clock is used, and both edges of the clock signal are used for an effective time resolution of 8.3 nanoseconds. This translates into a position resolution better than 0.001 inch. Note: It is possible to achieve higher resolutions with PWM transducers that are configured to perform multiple internal measurements (recirculations) and report the sum of those measurements in the pulse width.
Servo Loop Configuration	AXIS_SERVO AXIS_SERVO_DRIVE	INT	GSV SSV	The Servo Loop Configuration attribute determines the specific configuration of the servo loop topology when the axis is set to "servo". 0 = custom 1 = feedback only 2 = aux. feedback only 3 = position servo 4 = aux. position servo 5 = dual position servo 6 = dual command servo 7 = aux. dual command servo 8 = velocity servo 9 = torque servo 10 = dual command/feedback servo

Attribute	Axis Type	Data Type	Access	Description
Servo Output Level	AXIS_SERVO	REAL	GSV Tag	**Important**: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. Servo Output Level in Volts Servo Output Level is the current voltage level of the servo output of the specified axis. The Servo Output Level can be used in drilling applications, for example, where the servo module is interfaced to an external Torque Loop Servo Drive, to detect when the drill bit has engaged the surface of the work piece.
Servo Polarity Bits	AXIS_SERVO	DINT	GSV	0 = Feedback Polarity Negative 1 = Servo Polarity Negative **Feedback Polarity Negative** This Feedback Polarity Negative bit attribute controls the polarity of the encoder feedback and, when properly configured, insures that when the axis is moved in the user defined positive direction that the axis Actual Position value increases. This bit can be configured automatically using the MRHD and MAHD motion instructions. **Servo Polarity Negative** This Servo Polarity Negative bit attribute controls the polarity of the servo output to the drive. When properly configured along with the Feedback Polarity Negative bit, it insures that when the axis servo loop is closed that it is closed as a negative feedback system and not an unstable positive feedback system. This bit can be configured automatically using the MRHD and MAHD motion instructions.
Servo Status	AXIS_SERVO	DINT	Tag	Lets you access the status bits for your servo loop in one 32-bit word. This tag is the same as the Servo Status Bits attribute.

Servo Status	Bit
Servo Action Status	0
Drive Enable Status	1
Shutdown Status	2
Process Status	3
Output Limit Status	4
Position Lock Status	5
Home Input Status	6
Reg 1 Input Status	7
Reg 2 Input Status	8
Resevered	9
Resevered	10
Drive Fault Input Status	11

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Servo Status Bits</td><td>AXIS_SERVO</td><td>DINT</td><td>GSV*</td><td>Lets you access the status bits for your servo loop in one 32-bit word. This attribute is the same as the Servo Status tag.
<table><tr><th>Servo Status</th><th>Bit</th></tr><tr><td>Servo Action Status</td><td>0</td></tr><tr><td>Drive Enable Status</td><td>1</td></tr><tr><td>Shutdown Status</td><td>2</td></tr><tr><td>Process Status</td><td>3</td></tr><tr><td>Output Limit Status</td><td>4</td></tr><tr><td>Position Lock Status</td><td>5</td></tr><tr><td>Home Input Status</td><td>6</td></tr><tr><td>Reg 1 Input Status</td><td>7</td></tr><tr><td>Reg 2 Input Status</td><td>8</td></tr><tr><td>Resevered</td><td>9</td></tr><tr><td>Resevered</td><td>10</td></tr><tr><td>Drive Fault Input Status</td><td>11</td></tr></table></td></tr>
<tr><td>Shutdown Status</td><td>AXIS_CONSUMED
AXIS_GENERIC
AXIS_SERVO
AXIS_SERVO_DRIVE
AXIS_VIRTUAL</td><td>BOOL</td><td>Tag</td><td>If this bit is:
· ON — The axis is in the Shutdown state.
· OFF — The axis isn't in the Shutdown state.</td></tr>
<tr><td>Soft Overtravel Fault Action</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>SINT</td><td>GSV
SSV</td><td><table><tr><th>Fault Action</th><th>Value</th></tr><tr><td>Shutdown</td><td>0</td></tr><tr><td>Disable Drive</td><td>1</td></tr><tr><td>Stop Motion</td><td>2</td></tr><tr><td>Status Only</td><td>3</td></tr></table></td></tr>
<tr><td>SSI Clock Frequency</td><td>AXIS_SERVO</td><td>SINT</td><td>GSV</td><td>0 = 208 kHz
1 = 650 kHz
This attribute provides for setting the Clock Frequency in kHz of the SSI device. This attribute is only active if the Transducer Type is set to SSI.</td></tr>
<tr><td>SSI Code Type</td><td>AXIS_SERVO</td><td>SINT</td><td>GSV</td><td>0 = Binary
1 = Gray
This attribute provides for setting the whether the SSI device is using Binary or Gray code. This attribute is only active if the Transducer Type is set to SSI.</td></tr>
<tr><td>SSI Data Length</td><td>AXIS_SERVO</td><td>SINT</td><td>GSV</td><td>This attribute provides for setting the data length of the SSI device. This attribute is only active if the Transducer Type is set to SSI.</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Start Actual Position	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	Start Actual Position in Position Units Whenever a new motion planner instruction starts for an axis (for example, using a MAM instruction), the value of the axis command position and actual position is stored at the precise instant the motion begins. These values are stored as the Start Command Position and Start Actual Position respectively in the configured Position Units of the axis. Start Positions are useful to correct for any motion occurring between the *detection* of an event and the action *initiated* by the event. For instance, in coil winding applications, Start Command Positions can be used in an expression to compensate for overshooting the end of the bobbin before the gearing direction is reversed. If you know the position of the coil when the gearing direction was *supposed* to change, and the position at which it *actually* changed (the Start Command Position), you can calculate the amount of overshoot, and use it to correct the position of the wire guide relative to the bobbin.
Start Command Position	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	Start Command Position in Position Units Whenever a new motion planner instruction starts for an axis (for example, using a MAM instruction), the value of the axis command position and actual position is stored at the precise instant the motion begins. These values are stored as the Start Command Position and Start Actual Position respectively in the configured Position Units of the axis. Start Positions are useful to correct for any motion occurring between the *detection* of an event and the action *initiated* by the event. For instance, in coil winding applications, Start Command Positions can be used in an expression to compensate for overshooting the end of the bobbin before the gearing direction is reversed. If you know the position of the coil when the gearing direction was *supposed* to change, and the position at which it *actually* changed (the Start Command Position), you can calculate the amount of overshoot, and use it to correct the position of the wire guide relative to the bobbin.
Start Master Offset	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	Start Master Offset in Master Position Units The Start Master Offset is the position offset that was applied to the master side of the position cam when the last Motion Axis Move (MAM) instruction with the move type set to "Absolute Master Offset" or "Incremental Master Offset" was executed. The Start Master Offset is returned in master position units. The Start Master Offset will show the same unwind characteristic as the position of a linear axis.
Stopping Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set if there is a stopping process currently in progress. Cleared when the stopping process is complete. The stopping process is used to stop an axis (initiated by an MAS, MGS, Stop Motion fault action, or mode change).
Stopping Time Limit	AXIS_SERVO_DRIVE	REAL	GSV SSV	Sec This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually don't have to change it.

Attribute	Axis Type	Data Type	Access	Description
Stopping Torque	AXIS_SERVO_DRIVE	REAL	GSV SSV	% Rated This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually don't have to change it.
Strobe Actual Position	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	Strobe Actual Position in Position Units Strobe Actual Position, and Strobe Command Position are used to simultaneously store a snap-shot of the actual, command position and master offset position of an axis when the MGSP (Motion Group Strobe Position) instruction is executed. The values are stored in the configured Position Units of the axis. Since the MGSP instruction simultaneously stores the actual and command positions for all axes in the specified group of axes, the resultant Strobe Actual Position and Strobe Command Position values for different axes can be used to perform real time calculations. For example, the Strobe Actual Positions can be compared between two axis to provide a form of "slip compensation" in web handling applications.
Strobe Command Position	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	Strobe Command Position in Position Units Strobe Actual Position, and Strobe Command Position are used to simultaneously store a snap-shot of the actual, command position and master offset position of an axis when the MGSP (Motion Group Strobe Position) instruction is executed. The values are stored in the configured Position Units of the axis. Since the MGSP instruction simultaneously stores the actual and command positions for all axes in the specified group of axes, the resultant Strobe Actual Position and Strobe Command Position values for different axes can be used to perform real time calculations. For example, the Strobe Actual Positions can be compared between two axis to provide a form of "slip compensation" in web handling applications.
Strobe Master Offset	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	Strobe Master Offset in Master Position Units The Strobe Master Offset is the position offset that was applied to the master side of the position cam when the last Motion Group Strobe Position (MGSP) instruction was executed. The Strobe Master Offset is returned in master position units. The Strobe Master Offset will show the same unwind characteristic as the position of a linear axis.
Telegram Type	AXIS_SERVO_DRIVE	INT	GSV	Set to a value of 7, which means Application Telegram. See IDN 15 in IEC 1491.

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Test Direction Forward</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>SINT</td><td>GSV</td><td>The direction of axis travel during the last hookup test initiated by a MRHD (Motion Run Hookup Test) instruction.
0 = reverse
1 = forward (positive)

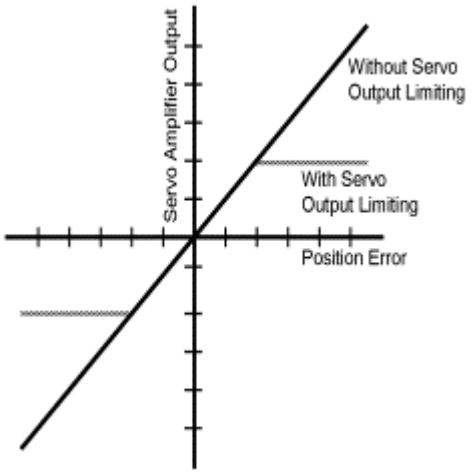
<table><tr><th>For this Data type</th><th>Details</th></tr><tr><td>AXIS_SERVO</td><td>This value doesn't depend on the Servo Polarity Bits attribute. The MAHD (Motion Apply Hookup Test) instruction uses the Test Direction Forward attribute and the Test Output Polarity attribute to set the Servo Polarity Bits attribute for negative feedback and correct directional sense.</td></tr><tr><td>AXIS_SERVO_DRIVE</td><td>This value doesn't depend on the Drive Polarity attribute. The MAHD (Motion Apply Hookup Test) instruction uses the Test Direction Forward attribute and the Test Output Polarity attribute to set the Drive Polarity attribute for the correct directional sense.</td></tr></table></td></tr>
<tr><td>Test Increment</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
SSV</td><td>Position Units

The Motor Feedback Test Increment attribute is used in conjunction with the MRHD (Motion Run Hookup Diagnostic) instruction to determine the amount of motion that is necessary to satisfy the MRHD initiated test process. This value is typically set to approximately a quarter of a revolution of the motor..</td></tr>
<tr><td>Test Status</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>INT</td><td>GSV</td><td>0 = test process successful
1 = test in progress
2 = test process aborted by user
3 = test process time-out fault (~2 seconds)
4 = test failed – servo fault
5 = test failed – insufficient test increment

More for AXIS_SERVO_DRIVE data type…
6 = test failed – wrong polarity
7 = test failed – missing signal
8 = test failed – device comm error
9 = test failed – feedback config error
10 = test failed – motor wiring error

This attribute returns the status of the last run MRHD (Motion Run Hookup Diagnostic) instruction that initiates a hookup diagnostic process on the axis. Use this attribute to determine when the MRHD initiated operation has successfully completed. Conditions may occur, however, that make it impossible to properly perform the operation. When that happens, the test process is automatically aborted and a test fault reported that is stored in the Test Status output parameter.</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Time Cam Pending Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set if a Time Cam motion profile is currently pending the completion of a currently executing cam profile. This would be initiated by executing an MATC instruction with Pending execution selected. This bit is cleared when the current time cam profile completes, initiating the start of the pending cam profile. This bit is also cleared if the time cam profile completes, or is superseded by some other motion operation.
Time Cam Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set if a Time Cam motion profile is currently in progress. Cleared when the Time Cam is complete or is superseded by some other motion operation.
Timer Event Fault	AXIS_SERVO AXIS_SERVO_DRIVE	BOOL	Tag	If this bit is set, the motion module has a problem with its timer event that synchronizes the module's servo loop to the master timebase of the chassis (that is, Coordinated System Time). To clear this bit, reconfigure the motion module.
Torque Command	AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important**: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. %Rated The command when operating in Torque Mode in terms of % rated.
Torque Data Scaling	AXIS_SERVO_DRIVE	INT	GSV	This attribute is derived from the Drive Units attribute. See IDN 86 in IEC 1491.
Torque Data Scaling Exp	AXIS_SERVO_DRIVE	INT	GSV	This attribute is derived from the Drive Units attribute. See IDN 94 in IEC 1491.
Torque Data Scaling Factor	AXIS_SERVO_DRIVE	DINT	GSV	This attribute is derived from the Drive Units attribute. See IDN 93 in IEC 1491.
Torque Feedback	AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important**: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. %Rated The torque feedback when operating in Torque Mode in terms of % rated.

Attribute	Axis Type	Data Type	Access	Description
Torque Limit Bipolar	AXIS_SERVO_DRIVE	REAL	GSV SSV	%Rated The Torque Limit attribute provides a method of limiting the maximum command current/torque to the motor to a specified level in terms of the motor's continuous current/torque rating. The output of the servo drive to the motor as a function of position servo error, both with and without servo torque limiting, is shown below.  The torque limit specifies the maximum percentage of the motors rated current that the drive can command as either positive or negative torque. For example, a torque limit of 150% shall limit the current delivered to the motor to 1.5 times the continuous current rating of the motor.
Torque Limit Negative	AXIS_SERVO_DRIVE	REAL	GSV SSV	%Rated This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually don't have to change it.
Torque Limit Positive	AXIS_SERVO_DRIVE	REAL	GSV SSV	%Rated This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually don't have to change it.

Attribute	Axis Type	Data Type	Access	Description
Torque Limit Source	AXIS_SERVO_DRIVE	DINT	GSV Tag	**Important**: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. This parameter displays the present source (if any) of any torque limiting for the axis. 0 = Not Limited 1 = Neg.e Torque Limit 2 = Pos. Torque Limit 3 = Amp Peak Limit 4 = Amp I(t) Limit 5 = Bus Regulator Limit 6 = Bipolar Torque Limit 7 = Motor Peak Limit 8 = Motor I(t) Limit 9 = Voltage Limit
Torque Limit Status	AXIS_SERVO_DRIVE	BOOL	Tag	Set when the magnitude of the axis torque command is greater than the configured Torque Limit.
Torque Offset	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV Tag	Torque Offset from –100% to +100% Torque Offset compensation can be used to provide a dynamic torque command correction to the output of the velocity servo loop. Since this value is updated synchronously every Coarse Update Period, the Torque Offset can be tied into custom outer control loop algorithms using Function Block programming.
Torque Polarity	AXIS_SERVO_DRIVE	INT	GSV	This attribute is derived from the Drive Polarity attribute. See IDN 85 in IEC 1491.

Attribute	Axis Type	Data Type	Access	Description
Torque Scaling	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	% / Position Units Per Second^2 The Torque Scaling attribute is used to convert the acceleration of the servo loop into equivalent % rated torque to the motor. This has the effect of "normalizing" the units of the servo loop's gain parameters so that their values are not affected by variations in feedback resolution, drive scaling, motor and load inertia, and mechanical gear ratios. In fact, the Torque Scaling value, when properly established, represents the inertia of the system and is related to the Tune Inertia attribute value by a factor of the Conversion Constant. · AXIS_SERVO — The Torque Scaling value is typically established by the MAAT instruction as part of the controller's automatic tuning procedure. · AXIS_SERVO_DRIVE — The Torque Scaling value is typically established by the drive's automatic tuning procedure. The value can be manually calculated, if necessary, using the following guidelines. Torque Scaling = 100% Rated Torque / (Acceleration @ 100% Rated Torque) For example, if this axis is using position units of motor revolutions (revs), and that with 100% rated torque applied to the motor, the motor accelerates at a rate of 3000 Revs/Sec^2, the Torque Scaling attribute value would be calculated as shown below. Torque Scaling = 100% Rated / (3000 RPS^2) = 0.0333% Rated/ Revs Per Second^2 Note that if the Torque Scaling value does not reflect the true torque to acceleration characteristic of the system, the gains also do not reflect the true performance of the system.
Torque Threshold	AXIS_SERVO_DRIVE	REAL	GSV SSV	%Rated This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually don't have to change it.
Torque Threshold Status	AXIS_SERVO_DRIVE	BOOL	Tag	Set when the magnitude of the physical axis Torque Feedback is less than the configured Torque Threshold.
Transform State Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	If the bit is: · ON — The axis is part of an active transform. · OFF — The axis isn't part of an active transform.

Attribute	Axis Type	Data Type	Access	Description
Tune Acceleration	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV	Position Units / Sec2 The Tune Acceleration and Tune Deceleration attributes return the measured acceleration and deceleration values for the last run tuning procedure. These values are used, in the case of an external torque servo drive configuration, to calculate the Tune Inertia value of the axis, and are also typically used by a subsequent MAAT (Motion Apply Axis Tune) to determine the tuned values for the Maximum Acceleration and Maximum Deceleration attributes.
Tune Acceleration Time	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV	Sec The Tune Acceleration Time and Tune Deceleration Time attributes return acceleration and deceleration time in seconds for the last run tuning procedure. These values are used to calculate the Tune Acceleration and Tune Deceleration attributes.
Tune Deceleration	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV	Position Units / Sec2 The Tune Acceleration and Tune Deceleration attributes return the measured acceleration and deceleration values for the last run tuning procedure. These values are used, in the case of an external torque servo drive configuration, to calculate the Tune Inertia value of the axis, and are also typically used by a subsequent MAAT (Motion Apply Axis Tune) to determine the tuned values for the Maximum Acceleration and Maximum Deceleration attributes.
Tune Deceleration Time	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV	Sec The Tune Acceleration Time and Tune Deceleration Time attributes return acceleration and deceleration time in seconds for the last run tuning procedure. These values are used to calculate the Tune Acceleration and Tune Deceleration attributes.

Attribute	Axis Type	Data Type	Access	Description
Tune Inertia	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV	% / MegaCounts Per Sec^2 The Tune Inertia value represents the total inertia for the axis as calculated from the measurements made during the tuning procedure. In actuality, the units of Tune Inertia are not industry standard inertia units but rather in terms of percent (%) of rated drive output per MegaCounts/Sec^2 of feedback input. In this sense it represents the input gain of torque servo drive. These units represent a more useful description of the inertia of the system as seen by the servo controller. The Tune Inertia value is used by the MAAT (Motion Apply Axis Tune) instruction to calculate the Torque Scaling. If the Tune Inertia value exceeds 100%Rated/MegaCounts Per $Second^2$, performance of the digital servo loop may be compromised due to excessive digitization noise associated with the velocity estimator. This noise is amplified by the Torque Scaling gain which is related to the Tune Inertia factor and passed on to the torque output of the drive. A high Tune Inertia value can, thus, result in excitation of mechanical resonances and also result in excessive heating of the motor due to high torque ripple. The only solution to this problem is to lower the loop bandwidths and optionally apply some output filtering. Since the Tune Inertia value represents a measure of the true system inertia, this situation can occur when driving a high inertia load relative to the motor, that is, a high inertia mismatch. But it can also occur when working with a drive that is undersized for the motor or with a system having low feedback resolution. In general, the lower the Tune Inertia the better the performance of the digital servo loops approximates that of an analog servo system. The product of the Tune Inertia (% Rated/MCPS) and the Velocity Servo BW (Hertz) can be calculated to directly determine quantization noise levels. Based on this product, the tuning algorithm can take action to limit high frequency noise injection to the motor. For motors with a Tune Inertia BW product of 1000 or more, the LP Filter is applied with a Filter BW of 5x the Velocity Servo Bandwidth in Hertz. This limits the amount of phase lag introduced by the LP filter to ~12 degrees which is relatively small compared to the 30 to 60 degrees of phase margin that we have for a typical tuned servo system. With a typical tuned LP filter BW value of 200 Hz, we can expect the high frequency quantization noise in the 1 KHz range to be attenuated roughly by a factor of 5. When the Tune Inertia BW product reaches 4000 or more, the LP filter alone is not going to be enough to manage the quantization noise level. So the tune algorithm begins to taper the system bandwidth by the ratio of 4000/(Tune Inertia * Vel Servo BW). This holds the quantization noise level at a fixed value, independent of the Tune Inertia BW product. For example, a motor with a Tune Inertia value of 213 and a Vel Servo BW of 41 Hz (8733 Inertia BW product) tunes with a Pos P Gain of 46 and a Vel P Gain of 117 and LP Filter BW of 93. This is a good noise-free gain set.

Attribute	Axis Type	Data Type	Access	Description
Tune Rise Time	AXIS_SERVO	REAL	GSV	Sec The Tune Rise Time attribute returns the axis rise time as measured during the tuning procedure. This value is only applicable to axes configured for interface to an external velocity servo drive. In this case, the Tune Rise Time attribute value is used to calculate the Tune Velocity Bandwidth.
Tune Speed Scaling	AXIS_SERVO	REAL	GSV	% / KiloCounts Per Sec The Tune Speed Scaling attribute returns the axis drive scaling factor measured during the tuning procedure. This value is only applicable to axes configured for interface to an external velocity servo drive. In this case, the Tune Speed Scaling attribute value is directly applied to the Velocity Scaling attribute by a subsequent MAAT (Motion Apply Axis Tune) instruction.
Tune Status	AXIS_SERVO AXIS_SERVO_DRIVE	INT	GSV	0 = tune process successful 1 = tune in progress 2 = tune process aborted by user 3 = tune process timed out 4 = • AXIS_SERVO — tune process failed due to servo fault • AXIS_SERVO_DRIVE — tune process failed due to drive fault 5 = axis reached Tuning Travel Limit 6 = axis polarity set incorrectly **More codes for a AXIS_SERVO_DRIVE…** 7 = tune measurement fault 8 = tune configuration fault The Tune Status attribute returns status of the last run MRAT (Motion Run Axis Tuning) instruction that initiates a tuning procedure on the targeted axis. Use the attribute to determine when the MRAT initiated operation has successfully completed. Conditions may occur, however, that make it impossible for the control to properly perform the operation. When this is the case, the tune process is automatically aborted and a tune fault reported that is stored in the Tune Status output parameter.

Attribute	Axis Type	Data Type	Access	Description
Tuning Configuration Bits	AXIS_SERVO AXIS_SERVO_DRIVE	DINT	GSV SSV	Bits 0 = Tuning Direction Reverse 1 = Tune Position Error Integrator 2 = Tune Velocity Error Integrator 3 = Tune Velocity Feedforward 4 = Tune Acceleration Feedforward 5 = Tune Output Low-Pass Filter 6 = bidirectional Tuning 7 = Tune Friction Compensation 8 = Tune Torque Offset **Tuning Direction Reverse** The Tune Direction Reverse bit determines the direction of the tuning procedure. If this bit is set (true), motion is initiated in the reverse (or negative) direction. **Tune Position Error Integrator** If this bit is: · ON — The tuning procedure calculates the Position Integral Gain. · OFF — The tuning procedure sets the Position Integral Gain to 0. **Tune Velocity Error Integrator** If this bit is: · ON — The tuning procedure calculates the Velocity Integral Gain. · OFF — The tuning procedure sets the Velocity Integral Gain to 0. **Tune Velocity Feedforward** If this bit is: · ON — The tuning procedure calculates the Velocity Feedforward Gain. · OFF — The tuning procedure sets the Velocity Feedforward Gain to 0. **Tune Acceleration Feedforward** If this bit is: · ON — The tuning procedure calculates the Acceleration Feedforward Gain. · OFF — The tuning procedure sets the Acceleration Feedforward Gain to 0. **Tune Output Low-Pass Filter** If this bit is: · ON — The tuning procedure calculates the Output Filter Bandwidth. · OFF — The tuning procedure sets the Output Filter Bandwidth to 0, which disables the filter. ***Continued on next page***

Attribute	Axis Type	Data Type	Access	Description
Tuning Configuration Bits (cont.)				**Bidirectional Tuning** The Bidirectional Tuning bit determines whether the tuning procedure is unidirectional or bidirectional. If this bit is set (true), the tuning motion profile is first initiated in the specified tuning direction and then is repeated in the opposite direction. Information returned by the Bidirectional Tuning profile can be used to tune Friction Compensation and Torque Offset. When configured for a hydraulics External Drive Type the bidirectional tuning algorithm also computes the Directional Scaling Ratio. **Tune Friction Compensation** This tuning configuration is only valid if configured for bidirectional tuning. If this bit is: · ON — The tuning procedure calculates the Friction Compensation Gain. · OFF — The Friction Compensation Gain is not affected. **Tune Torque Offset** This tuning configuration is only valid if configured for bidirectional tuning. If this bit is: · ON — The tuning procedure calculates the Torque Offset. · OFF — The Torque Offset is not affected.
Tuning Speed	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units / Sec The Tuning Speed attribute sets the maximum speed of the tuning procedure. This attribute should be set to the desired maximum operating speed of the motor before you run the tuning procedure. The tuning procedure measures maximum acceleration and deceleration rates based on ramps to and from the Tuning Speed. Thus, the accuracy of the measured acceleration and deceleration capability is reduced by tuning at a speed other than the desired operating speed of the system..
Tuning Torque	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	% The Tuning Torque attribute determines the maximum torque of the tuning procedure. This attribute should be set to the desired maximum safe torque level before you run the tuning procedure. The default value is 100%, which yields the most accurate measure of the acceleration and deceleration capabilities of the system. In some cases a lower tuning torque limit value may be desirable to limit the stress on the mechanics during the tuning procedure. In this case the acceleration and deceleration capabilities of the system are extrapolated based on the ratio of the tuning torque to the maximum torque output of the system. Note that the extrapolation error increases as the Tuning Torque value decreases.

Attribute	Axis Type	Data Type	Access	Description
Tuning Travel Limit	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units The Tuning Travel Limit attribute limits the travel of the axis during the tuning procedrue. If the axis can't complete the tuning procedure before exceeding the Tuning Travel Limit, the motion module stops the tuning procedure and reports that the Tuning Travel Limit was exceeded via the Tune Status attribute. This does not mean that the Tuning Travel Limit was actually exceeded, but that had the tuning procedure gone to completion that the limit would have been exceeded.
Velocity Command	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important**: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. Velocity Command in Position Units / Sec Velocity Command is the current velocity reference to the velocity servo loop, in the configured axis Position Units per Second, for the specified axis. The Velocity Command value, hence, represents the output of the outer position control loop. Velocity Command is not to be confused with Command Velocity, which represents the rate of change of Command Position input to the position servo loop.
Velocity Data Scaling	AXIS_SERVO_DRIVE	INT	GSV	This attribute is derived from the Drive Units attribute. See IDN 44 in IEC 1491.
Velocity Data Scaling Exp	AXIS_SERVO_DRIVE	INT	GSV	This attribute is derived from the Drive Units attribute. See IDN 46 in IEC 1491.
Velocity Data Scaling Factor	AXIS_SERVO_DRIVE	DINT	GSV	This attribute is derived from the Drive Units attribute. See IDN 45 in IEC 1491.
Velocity Droop	AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units / sec This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually don't have to change it.
Velocity Error	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important**: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. Velocity Error in Position Units / Sec Velocity Error is the difference, in configured axis Position Units per Second, between the commanded and actual velocity of an axis. For an axis with an active velocity servo loop, velocity error is used, along with other error terms, to drive the motor to the condition where the velocity feedback is equal to the velocity command..

Attribute	Axis Type	Data Type	Access	Description
Velocity Feedback	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important**: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. Velocity Feedback in Position Units / Sec Velocity Feedback is the actual velocity of the axis as estimated by the motion module, in the configured axis Position Units per second. The estimated velocity is computed by applying a 1 KHz low-pass filter to the change in actual position over the servo update interval. Velocity Feedback is a signed value—the sign (+ or -) depends on which direction the axis is currently moving.
Velocity Feedforward Gain	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	% Servo Drives require non-zero command input to generate steady-state axis acceleration or velocity. To provide the non-zero output from the Servo Module a non-zero position or velocity error needs to be present. We call this dynamic error while moving "following error". Well, this non-zero following error condition is a situation we are trying to avoid. We ideally want zero following error -- all the time. This could be achieved through use of the position integral gain controls as described above, but typically the response time of the integrator action is too slow to be effective. An alternative approach that has superior dynamic response is to use Velocity and Acceleration Feedforward. The Velocity Feedforward Gain attribute is used to provide the Velocity Command output necessary to generate the commanded velocity. It does this by scaling the current Command Velocity by the Velocity Feedforward Gain and adding it as an offset to the Velocity Command generated by the position loop control elements. With this done, the position loop control elements do not need to generate much of a contribution to the Velocity Command, hence the Position Error value is significantly reduced. Hence, the Velocity Feedforward Gain allows the following error of the servo system to be reduced to nearly zero when running at a constant speed. This is important in applications such as electronic gearing and synchronization applications where it is necessary that the actual axis position not significantly lag behind the commanded position at any time. The optimal value for Velocity Feedforward Gain is 100% theoretically. In reality, however, the value may need to be tweaked to accommodate velocity loops with non-infinite loop gain and other application considerations. One thing that may force a smaller Velocity Feedforward value is that increasing amounts of feedforward tends to exacerbate axis overshoot. If necessary, the Velocity Feedforward Gain may be "tweaked" from the 100% value by running a simple user program that jogs the axis in the positive direction and monitor the Position Error of the axis during the jog. Increase the Velocity Feedforward Gain until the Position Error at constant speed is as small as possible, but still positive. If the Position Error at constant speed is negative, the actual position of the axis is *ahead* of the command position. If this occurs, decrease the Velocity Feedforward Gain such that the Position Error is again positive. Note that reasonable maximum velocity, acceleration, and deceleration values must be entered to jog the axis.

Attribute	Axis Type	Data Type	Access	Description
Velocity Integral Gain	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	1/mSec-Sec When configured for a torque (current) loop servo drive, every servo update the current Velocity Error is also accumulated in a variable called the Velocity Integral Error. This value is multiplied by the Velocity Integral Gain to produce a component to the Servo Output or Torque Command that attempts to correct for the velocity error. The characteristic of Vel I Gain correction, however, is that any non-zero Velocity Error accumulates in time to generate enough force to make the correction. This attribute of Vel I Gain makes it invaluable in applications where velocity accuracy is critical. The higher the Vel I Gain value the faster the axis is driven to the zero Velocity Error condition. Unfortunately, I Gain control is intrinsically unstable. Too much I Gain results in axis oscillation and servo instability. In certain cases, Vel I Gain control is disabled. One such case is when the servo output to the axis' drive is saturated. Continuing integral control behavior in this case would only exacerbate the situation. Another common case is when performing certain motion. When the Integrator Hold Enable attribute is set, the servo loop automatically disables the integrator during commanded motion. Due to the destabilizing nature of Integral Gain, it is recommended that Position Integral Gain and Velocity Integral Gain be considered mutually exclusive. If Integral Gain is needed for the application use one or the other, but not both. In general, where static positioning accuracy is required, Velocity Integral Gain is the better choice. The typical value for the Velocity Integral Gain is ~15 $mSec^{-1}$-Sec^{-1}. **If you have an AXIS_SERVO_DRIVE data type…** While the Vel I Gain, if employed, is typically established by the automatic servo tuning procedure, the Pos I Gain value may also be set manually. Before doing this it must be stressed that the Torque Scaling factor for the axis must be established for the drive system. Refer to Torque Scaling attribute description for an explanation of how the Torque Scaling factor can be calculated. Once this is done the Vel I Gain can be computed based on the current or computed value for the Vel P Gain using the following formula: $\text{Vel I Gain} = 0.25 * 0.001\ \text{Sec/mSec} * (\text{Vel P Gain})^2$ Assuming a Vel P Gain value of 0.25 Sec^{-1} this results in a Vel I Gain value of ~15.6 $mSec^{-1}$-Sec^{-1}-

Attribute	Axis Type	Data Type	Access	Description
Velocity Integrator Error	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV Tag	**Important**: To use this attribute, choose it as one of the attributes for Real Time Axis Information for the axis. Otherwise, you won't see the right value as the axis runs. See Axis Info Select 1. Velocity Integrator Error in Position Units – mSec / Sec Velocity Integrator Error is the running sum of the Velocity Error, in the configured axis Position Units per Second, for the specified axis. For an axis with an active velocity servo loop, the velocity integrator error is used, along with other error terms, to drive the motor to the condition where the velocity feedback is equal to the velocity command.
Velocity Limit Bipolar	AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units / sec This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually don't have to change it.
Velocity Limit Negative	AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units / sec This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually don't have to change it.
Velocity Limit Positive	AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units / sec This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually don't have to change it.
Velocity Limit Status	AXIS_SERVO_DRIVE	BOOL	Tag	Set when the magnitude of the commanded velocity to the velocity servo loop input is greater than the configured Velocity Limit.
Velocity Lock Status	AXIS_SERVO_DRIVE	BOOL	Tag	Set when the magnitude of the physical axis Velocity Feedback is within the configured Velocity Window of the current velocity command.
Velocity Offset	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV Tag	Velocity Offset in Position Units / Sec Velocity Offset compensation can be used to give a dynamic velocity correction to the output of the position servo loop. Since this value is updated synchronously every Coarse Update Period, the Velocity Offset can be tied into custom outer control loop algorithms using Function Block programming.
Velocity Polarity	AXIS_SERVO_DRIVE	INT	GSV	This attribute is derived from the Drive Polarity attribute. See IDN 42 in IEC 1491.

Attribute	Axis Type	Data Type	Access	Description
Velocity Proportional Gain	AXIS_SERVO AXIS_SERVO_DRIVE	REAL	GSV SSV	1/Sec **AXIS_SERVO** When configured for a torque (current) loop servo drive, the servo module's digital velocity loop provides damping without the requirement for an analog tachometer. The Velocity Error is multiplied by the Velocity Proportional Gain to produce a component to the Servo Output or Torque Command that ultimately attempts to correct for the velocity error, creating the damping effect. Thus, increasing the Velocity Proportional Gain results in smoother motion, enhanced acceleration, reduced overshoot, and greater system stability. The velocity loop also allows higher effective position loop gain values to be used, however, too much Velocity Proportional Gain leads to high frequency instability and resonance effects. Note that units for Velocity Proportional Gain are identical to that of the Position Proportional Gain making it easy to perform classic inches/min/mil calculations to determine static stiffness or damping. **Maximum Bandwidth** There are limitations to the maximum bandwidth that can be achieved for the velocity loop based on the dynamics of the torque loop of the servo drive and the desired damping of the system, Z. These limitations may be expressed as follows: Bandwidth (Velocity) = 0.25 * $1/Z^2$ * Bandwidth (Torque) For example, if the bandwidth of the drive's torque loop is 100 Hz and the damping factor, Z, is 0.8, the velocity bandwidth is approximately 40 Hz. Based on this number the corresponding gains for the loop can be computed. Note that the bandwidth of the torque loop includes feedback sampling delay and filter time constant. The velocity loop in the motion controller is not used when the servo module is configured for a velocity loop servo drive, Thus, establishing the Velocity Proportional Gain is not required in this case. The typical value for the Velocity Proportional Gain is ~250 Sec^{-1}. ***Continued on next page***

Attribute	Axis Type	Data Type	Access	Description
Velocity Proportional Gain (cont.)				**AXIS_SERVO_DRIVE** The standard RA SERCOS drive's digital velocity loop provides damping without the requirement for an analog tachometer. The Velocity Error is multiplied by the Velocity Proportional Gain to produce a Torque Command that ultimately attempts to correct for the velocity error, creating the damping effect. Thus, increasing the Velocity Proportional Gain results in smoother motion, enhanced acceleration, reduced overshoot, and greater system stability. The velocity loop also allows higher effective position loop gain values to be used, however, too much Velocity Proportional Gain leads to high frequency instability and resonance effects. Note that units for Velocity Proportional Gain are identical to that of the Position Proportional Gain making it easy to perform classic calculations to determine damping and bandwidth. If you know the desired unity gain bandwidth of the velocity servo in Hertz, use the following formula to calculate the corresponding P gain. Vel P Gain = Bandwidth (Hertz) / 6.28 In general, modern velocity servo systems typically run with a unity gain bandwidth of ~40 Hertz. The typical value for the Velocity Proportional Gain is ~250 Sec^{-1}. **Maximum Bandwidth** There are limitations to the maximum bandwidth that can be achieved for the velocity loop based on the dynamics of the inner torque loop of the system and the desired damping of the system, Z. These limitations may be expressed as follows: Bandwidth (Velocity) = 0.25 * $1/Z^2$ * Bandwidth (Torque) For example, if the bandwidth of the drive's torque loop is 100 Hz and the damping factor, Z, is 0.8, the velocity bandwidth is approximately 40 Hz. Based on this number the corresponding gains for the loop can be computed. Note that the bandwidth of the torque loop includes feedback sampling delay and filter time constant.

Attribute	Axis Type	Data Type	Access	Description
Velocity Scaling	AXIS_SERVO	REAL	GSV SSV	% / Position Units Per Second The Velocity Scaling attribute is used to convert the output of the servo loop into equivalent voltage to an external velocity servo drive. This has the effect of "normalizing" the units of the servo loop gain parameters so that their values are not affected by variations in feedback resolution, drive scaling, or mechanical gear ratios. The Velocity Scaling value is typically established by servo's automatic tuning procedure but these values can be calculated if necessary using the following guidelines. If the axis is using a velocity servo drive, the software velocity loop in the servo module is disabled. In this case the Velocity Scaling value can be calculated by the following formula: Velocity Scaling = 100% / (Speed @ 100%) For example, if this axis is using position units of motor revolutions (revs), and the servo drive is scaled such that with an input of 100% (for example, 10 Volts) the motor goes 5,000 RPM (or 83.3 RPS), the Torque Scaling attribute value would be calculated as shown below. Velocity Scaling = 100% / (83.3 RPS) = 1.2% / Revs Per Second

<table>
<tr><th>Attribute</th><th>Axis Type</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Velocity Servo Bandwidth</td><td>AXIS_SERVO
AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
SSV</td><td>Hertz

The value for the Velocity Servo Bandwidth represents the unity gain bandwidth that is to be used to calculate the gains for a subsequent MAAT (Motion Apply Axis Tune) instruction. The unity gain bandwidth is the frequency beyond which the velocity servo is unable to provide any significant position disturbance correction. In general, within the constraints of a stable servo system, the higher the Velocity Servo Bandwidth is the better the dynamic performance of the system. A maximum value for the Velocity Servo Bandwidth is generated by the MRAT (Motion Run Axis Tune) instruction. Computing gains based on this maximum value via the MAAT instruction results in dynamic response in keeping with the current value of the Damping Factor described above. Alternatively, the responsiveness of the system can be “softened” by reducing the value of the Velocity Servo Bandwidth before executing the MAAT instruction..

There are practical limitations to the maximum Velocity Servo Bandwidth for the velocity servo loop based on the drive system and, in some cases, the desired damping factor of the system, Z. Exceeding these limits could result in an unstable servo operation.

<table><tr><th>Data type</th><th>Bandwidth limits</th></tr><tr><td>AXIS_SERVO</td><td>For an external velocity loop servo drive,

Max Velocity Servo Bandwidth (Hz) = 0.159 * 2/Tune Rise Time

For an external torque loop servo drive,

Max Velocity Servo Bandwidth (Hz) = 0.159 * 0.25 * 1/Z^2 * 1/Drive Model Time Constant</td></tr><tr><td>AXIS_SERVO_DRIVE</td><td>Max Velocity Servo Bandwidth (Hz) = 0.159 * 0.25 * 1/Z^2 * 1/Drive Model Time Constant</td></tr></table>
The factor of 0.159 represents the 1/2PI factor required to convert Radians per Second units to Hertz.</td></tr>
<tr><td>Velocity Standstill Status</td><td>AXIS_SERVO_DRIVE</td><td>BOOL</td><td>Tag</td><td>Set when the magnitude of the physical axis Velocity Feedback is less than the configured Velocity Standstill Window.</td></tr>
<tr><td>Velocity Standstill Window</td><td>AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
SSV</td><td>Position Units / sec

This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually don’t have to change it.</td></tr>
<tr><td>Velocity Threshold</td><td>AXIS_SERVO_DRIVE</td><td>REAL</td><td>GSV
SSV</td><td>Position Units / sec

This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually don’t have to change it.</td></tr>
</table>

Attribute	Axis Type	Data Type	Access	Description
Velocity Threshold Status	AXIS_SERVO_DRIVE	BOOL	Tag	Set when the magnitude of the physical axis Velocity Feedback is less than the configured Velocity Threshold.
Velocity Window	AXIS_SERVO_DRIVE	REAL	GSV SSV	Position Units / sec This attribute maps directly to a SERCOS IDN. See the SERCOS Interface standard for a description. This attribute is automatically set. You usually don't have to change it.
Watch Event Armed Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set when a watch event has been armed through execution of the MAW (Motion Arm Watch) instruction. Cleared when either a watch event occurs or a MDW (Motion Disarm Watch) instruction is executed.
Watch Event Status	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	BOOL	Tag	Set when a watch event has occurred. Cleared when either another MAW (Motion Arm Watch) instruction or a MDW (Motion Disarm Watch) instruction is executed.
Watch Event Task	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	DINT	MSG	Shows which task is triggered when the watch event happens. · An instance of 0 means that no event task is configured to be triggered by the watch event. · The task is triggered at the same time that the Process Complete bit is set for the instruction that armed the watch event. · The controller sets this attribute. Don't set it by an external device.
Watch Position	AXIS_CONSUMED AXIS_GENERIC AXIS_SERVO AXIS_SERVO_DRIVE AXIS_VIRTUAL	REAL	GSV Tag	Watch Position in Position Units Watch Position is the current set-point position of an axis, in the configured axis Position Units, as set up in the last, most recently executed, MAW (Motion Arm Watch) instruction for that axis.

Additional Error Code Information

See these manuals for more information about error codes displayed on drives and/or multi-axis motion control systems.

Publication	Publication Number
Kinetix 2000 Multi-Axis Drive User Manual	2093-UM001
Kinetix 6000 Multi-Axis Drive User Manual	2094-UM001
Kinetix 7000 High Power Servo Drive User Manual	2099-UM001

Publication	Publication Number
Ultra 3000 Digital Servo Drive Installation Instructions	2098-IN003
8720 High Performance Drive Installation Instructions	8720MC-IN001
1394 SERCOS Interface Multi-Axis Motion Control System Installation Manual	1394-IN002

Notes:

Servo Loop Block Diagrams

Introduction

This appendix shows the servo loop block diagrams for common motion configurations.

Topic	Page
Interpreting the Diagrams	387
AXIS_SERVO	388
AXIS_SERVO_DRIVE	390

Interpreting the Diagrams

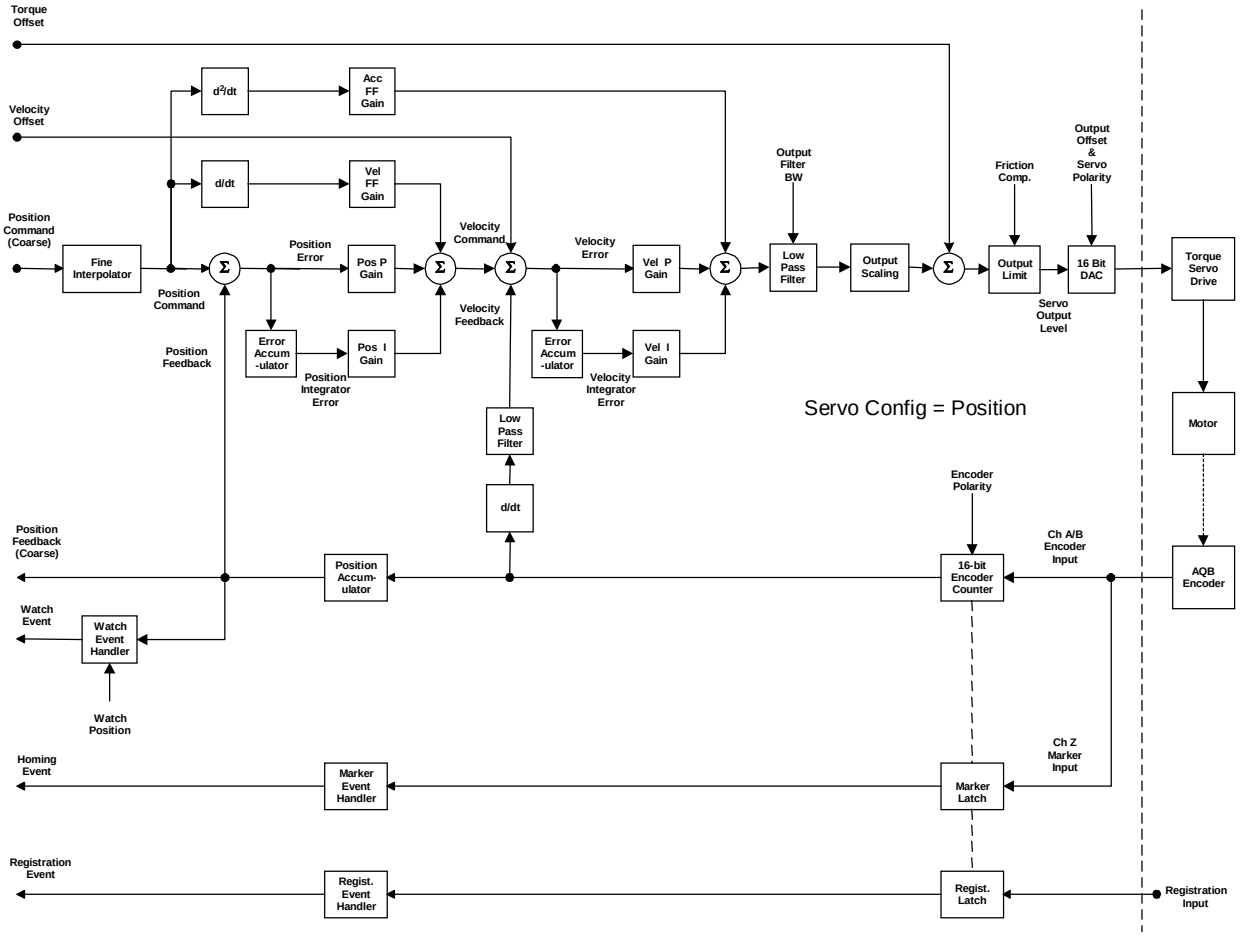
The diagrams use these labels for axes attributes.

Label	AXIS Attribute
Acc FF Gain	AccelerationFeedforwardGain
Friction Comp	FrictionCompensation
Output Filter BW	OutputFilterBandwidth
Output Limit	OutputLimit
Output Offset	OutputOffset
Output Scaling	OutputScaling
Pos I Gain	PositionIntegralGain
Pos P Gain	PositionProportionalGain
Position Error	PositionError
Position Integrator Error	PositionIntegratorError
Registration Position	RegistrationPosition
Servo Output Level	ServoOutputLevel
Vel FF Gain	VelocityFeedforwardGain
Vel I Gain	VelocityIntegralGain
Vel P Gain	VelocityProportionalGain
Velocity Command	VelocityCommand
Velocity Error	VelocityError
Velocity Feedback	VelocityFeedback
Velocity Integrator Error	VelocityIntegratorError
Watch Position	WatchPosition

AXIS_SERVO

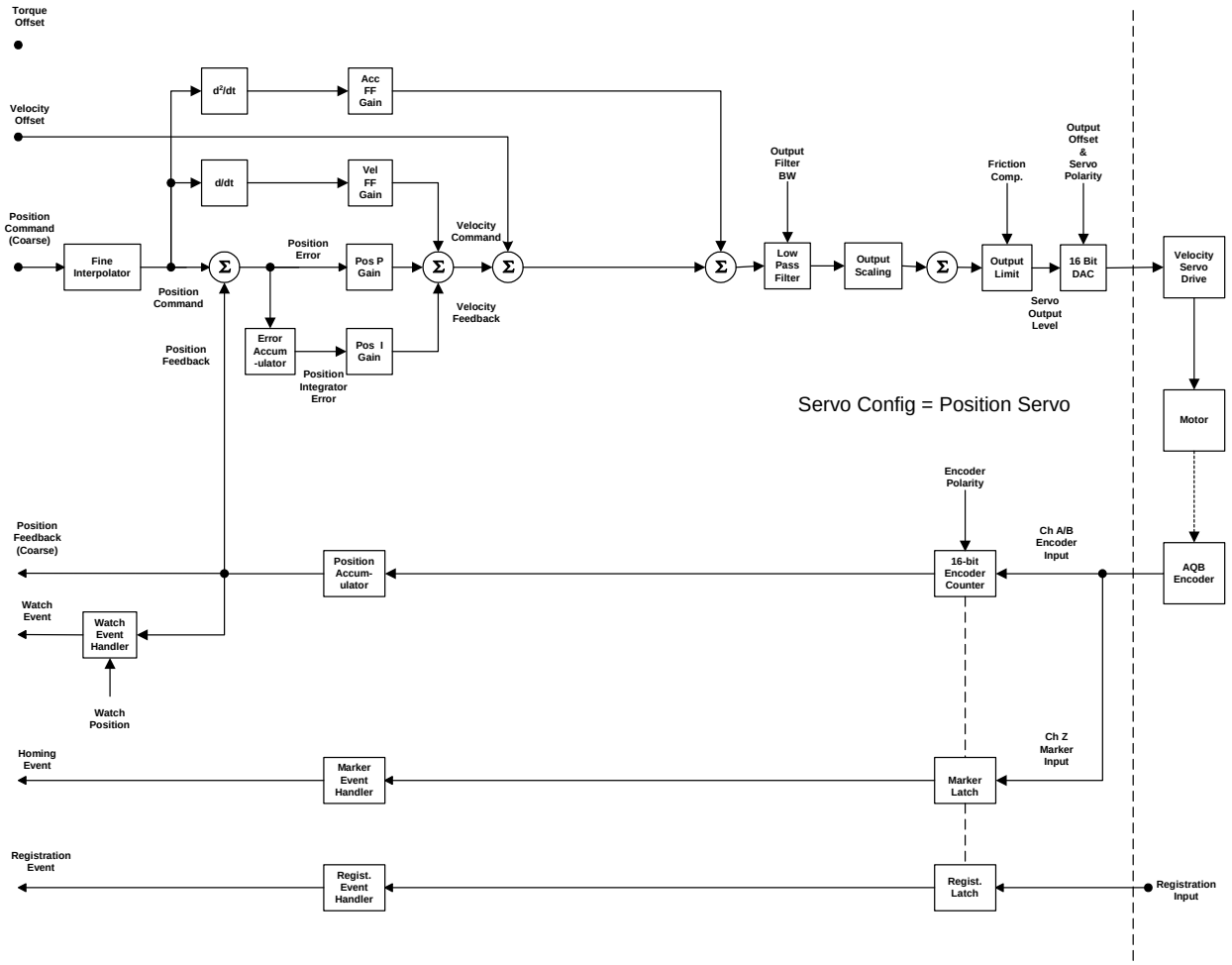
Topic	Page
Position Servo with Torque Servo Drive	388
Position Servo with Velocity Servo Drive	389

Position Servo with Torque Servo Drive



This configuration gives full position servo control using an external torque loop servo drive. Synchronous input data to the servo loop includes Position Command, Velocity Offset, and Torque Offset. The controller updates these values at the coarse update period of the motion group. The Position Command value is derived directly from the output of the motion planner, while the Velocity Offset and Torque Offset values are derived from the current value of the corresponding attributes.

Position Servo with Velocity Servo Drive

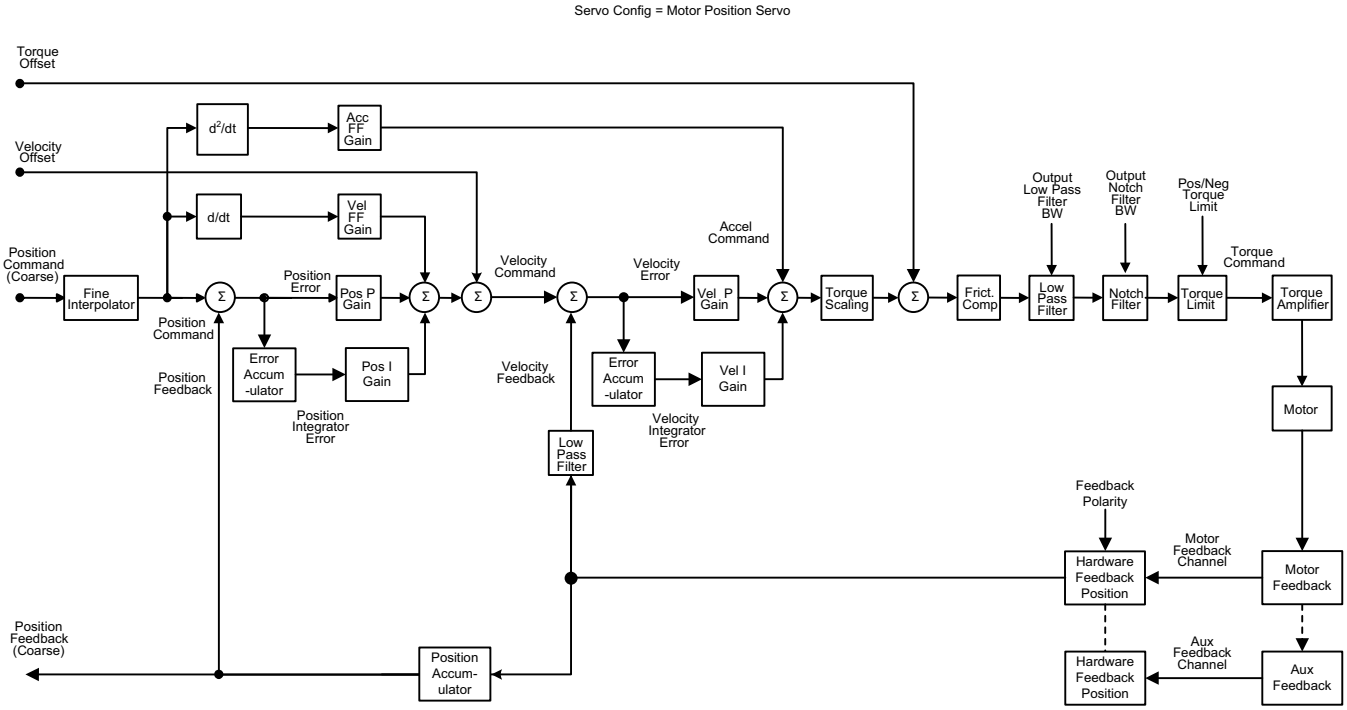


This configuration provides full position servo control using an external velocity loop servo drive. Note that in this configuration the servo module does not close the velocity loop, but rather the drive does. Synchronous input data to the servo loop includes Position Command and Velocity Offset. (Torque Offset is ignored.) The controller updates these values at the coarse update period of the motion group. The Position Command value is derived directly from the output of the motion planner, while the Velocity Offset value is derived from the current value of the corresponding attributes.

AXIS_SERVO_DRIVE

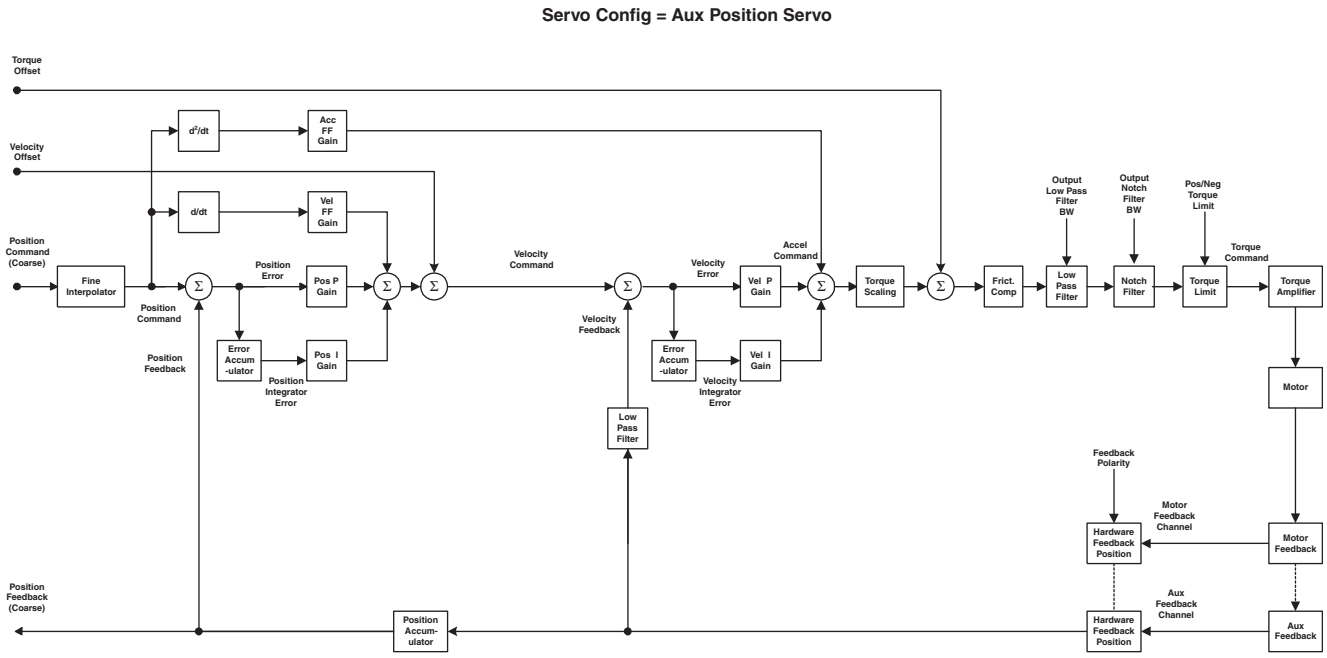
Topic	Page
Motor Position Servo	391
Auxiliary Position Servo	392
Dual Feedback Servo	393
Motor Dual Command Servo	394
Auxiliary Dual Command Servo	395
Dual Command Feedback Servo	395
Velocity Servo	396
Torque Servo	396
Drive Gains	397

Motor Position Servo



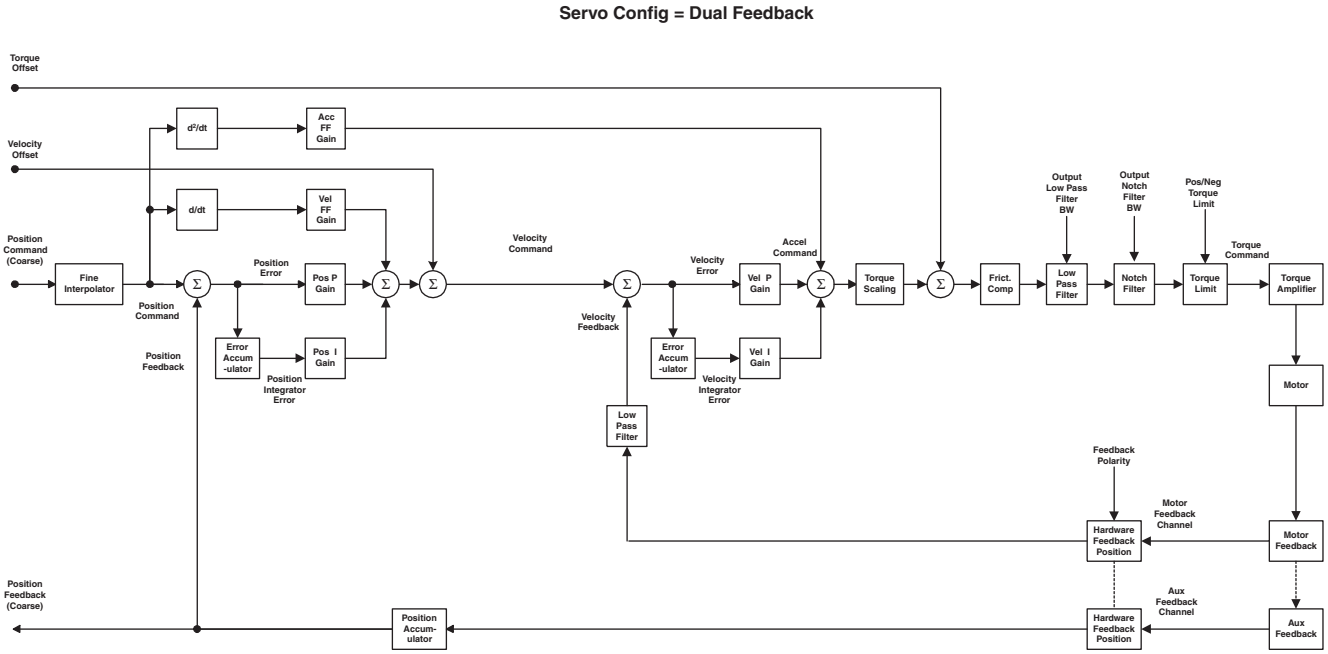
The Motor Position Servo configuration provides full position servo control using only the motor mounted feedback device to provide position and velocity feedback. This servo configuration is a good choice in applications where smoothness and stability are more important that positioning accuracy. Positioning accuracy is limited due to the fact that the controller has no way of compensating for non-linearity in the mechanics external to the motor. Note that the motor mounted feedback device also provides motor position information necessary for commutation. Synchronous input data to the servo loop includes Position Command, Velocity Offset, and Torque Offset. These values are updated at the coarse update rate of the associated motion group. The Position Command value is derived directly from the output of the motion planner, while the Velocity Offset and Torque Offset values are derived from the current value of the corresponding attributes. These offset attributes may be changed programmatically via SSV instructions or direct Tag access which, when used in conjunction with future Function Block programs, provides custom "outer" control loop capability.

Auxiliary Position Servo



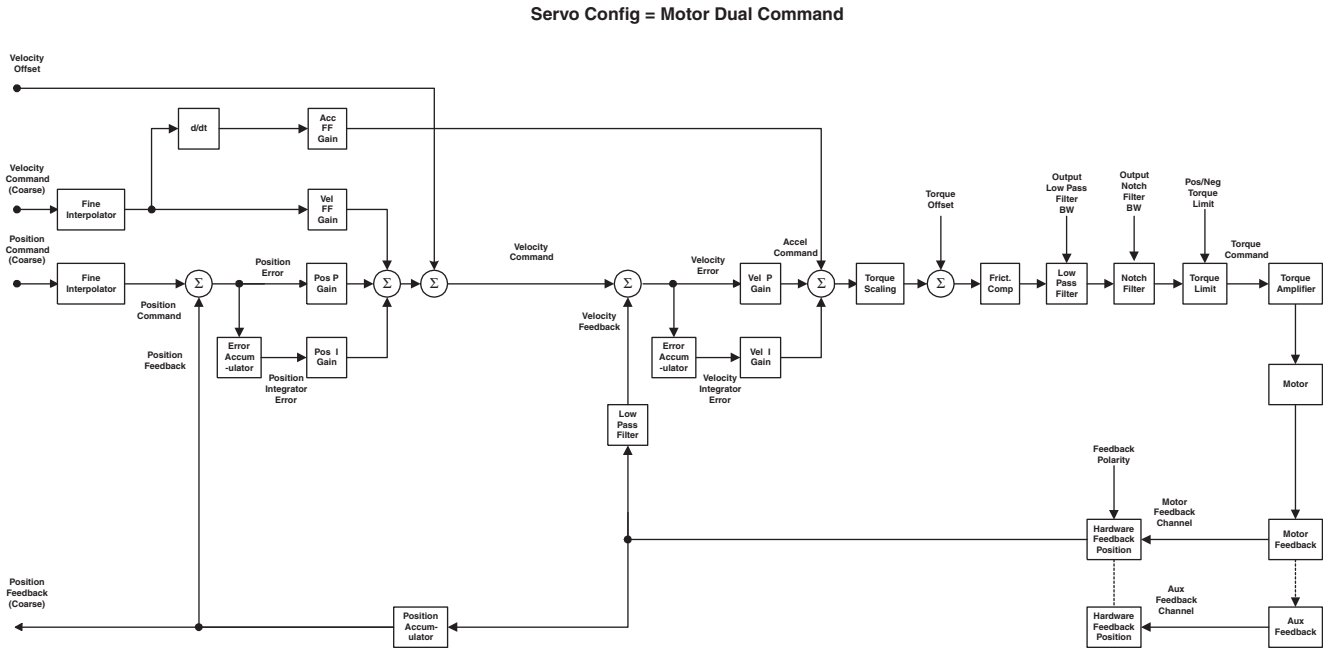
The Auxiliary Position Servo configuration provides full position servo control using an auxiliary (that is, external to the motor) feedback device to provide position and velocity feedback. This servo configuration is a good choice in applications positioning accuracy is important. The smoothness and stability may be limited, however, due to the mechanical non-linearities external to the motor. Note, that the motor mounted feedback device is still required to provide motor position information necessary for commutation. Synchronous input data to the servo loop includes Position Command, Velocity Offset, and Torque Offset. These values are updated at the coarse update rate of the associated motion group. The Position Command value is derived directly from the output of the motion planner, while the Velocity Offset and Torque Offset values are derived from the current value of the corresponding attributes. These offset attributes may be changed programmatically via SSV instructions or direct Tag access which, when used in conjunction with future Function Block programs, provides custom "outer" control loop capability.

Dual Feedback Servo



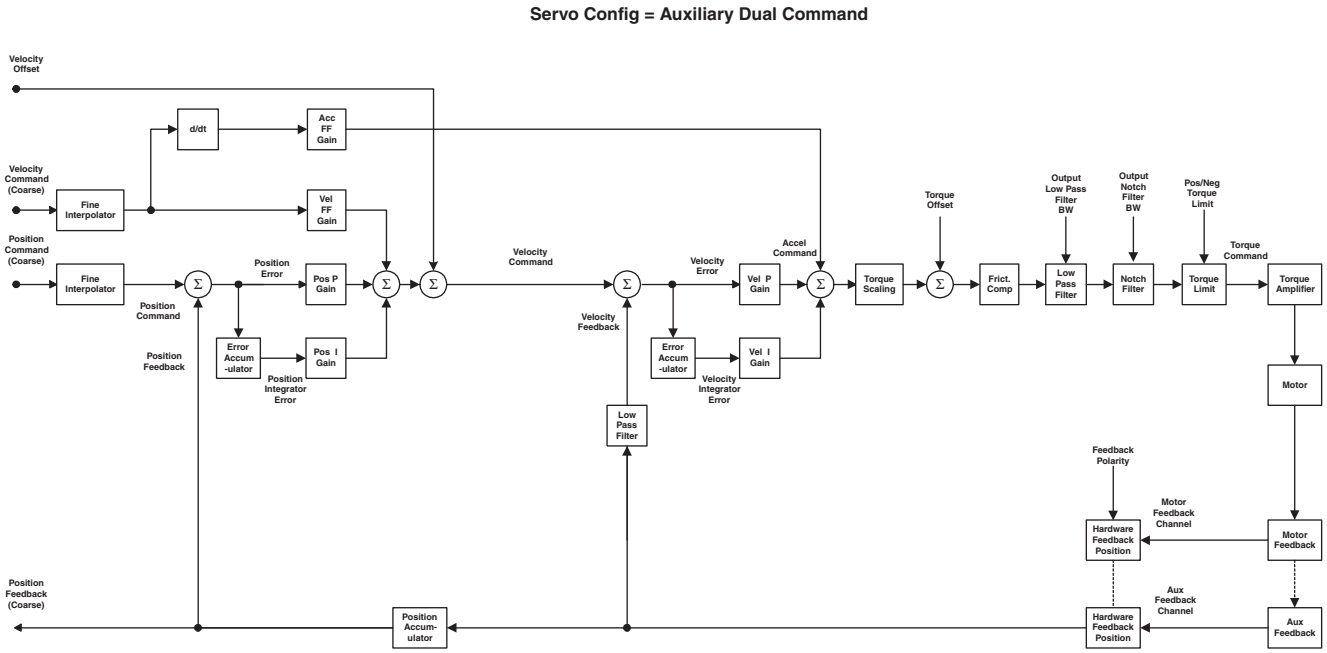
This configuration provides full position servo control using the auxiliary feedback device for position feedback and the motor mounted feedback device to provide velocity feedback. This servo configuration combines the advantages of accurate positioning associated with the auxiliary position servo with the smoothness and stability of the motor position servo configuration. Note that the motor mounted feedback device also provides motor position information necessary for commutation. Synchronous input data to the servo loop includes Position Command, Velocity Offset, and Torque Offset. These values are updated at the coarse update rate of the associated motion group. The Position Command value is derived directly from the output of the motion planner, while the Velocity Offset and Torque Offset values are derived from the current value of the corresponding attributes. These offset attributes may be changed programmatically via SSV instructions or direct Tag access which, when used in conjunction with future Function Block programs, provides custom "outer" control loop capability.

Motor Dual Command Servo



The Motor Dual Command Servo configuration provides full position servo control using only the motor mounted feedback device to provide position and velocity feedback. Unlike the Motor Position Servo configuration, however, both command position and command velocity are applied to the loop to provide smoother feedforward behavior. This servo configuration is a good choice in applications where smoothness and stability are important. Positioning accuracy is limited due to the fact that the controller has no way of compensating for non-linearities in the mechanics external to the motor. Note that the motor mounted feedback device also provides motor position information necessary for commutation. Synchronous input data to the servo loop includes Position Command, Velocity Command, and Velocity Offset. These values are updated at the coarse update rate of the associated motion group. The Position and Velocity Command values are derived directly from the output of the motion planner, while the Velocity Offset value is derived from the current value of the corresponding attributes. The velocity offset attribute may be changed programmatically via SSV instructions or direct Tag access which, when used in conjunction with future Function Block programs, provides custom "outer" control loop capability.

Auxiliary Dual Command Servo



The Auxiliary Dual Command Servo configuration provides full position servo control using only the auxiliary mounted feedback device to provide position and velocity feedback. Unlike the Auxiliary Position Servo configuration, however, both command position and command velocity are applied to the loop to provide smoother feedforward behavior. This servo configuration is a good choice in applications where positioning accuracy and good feedforward performance is important. The smoothness and stability may be limited, however, due to the mechanical non-linearities external to the motor. Note, that the motor mounted feedback device is still required to provide motor position information necessary for commutation. Synchronous input data to the servo loop includes Position Command, Velocity Command, and Velocity Offset. These values are updated at the coarse update rate of the associated motion group. The Position and Velocity Command values are derived directly from the output of the motion planner, while the Velocity Offset value is derived from the current value of the corresponding attributes. The velocity offset attribute may be changed programmatically via SSV instructions or direct Tag access which, when used in conjunction with future Function Block programs, provides custom "outer" control loop capability.

Dual Command Feedback Servo

The Motor Dual Command Feedback Servo configuration provides full position servo control using the auxiliary feedback device for position feedback and the motor mounted feedback device to provide velocity

feedback. Unlike the Dual Feedback Servo configuration, however, both command position and command velocity are also applied to the loop to provide smoother feedforward behavior. This servo configuration is a good choice in applications where smoothness and stability are important as well as positioning accuracy. Note, that the motor mounted feedback device is still required to provide motor position information necessary for commutation. Synchronous input data to the servo loop includes Position Command, Velocity Command, and Velocity Offset. These values are updated at the coarse update rate of the associated motion group. The Position and Velocity Command values are derived directly from the output of the motion planner, while the Velocity Offset value is derived from the current value of the corresponding attributes. The velocity offset attribute may be changed programmatically via SSV instructions or direct Tag access which, when used in conjunction with future Function Block programs, provides custom "outer" control loop capability.

Velocity Servo

The Velocity Servo configuration provides velocity servo control using the motor mounted feedback device. Synchronous input data to the servo loop includes Velocity Command, Velocity Offset, and Torque Offset. These values are updated at the coarse update rate of the associated motion group. The Velocity Command value is derived directly from the output of the motion planner, while the Velocity Offset and Torque Offset values are derived from the current value of the corresponding attributes. These offset attributes may be changed programmatically via SSV instructions or direct Tag access which, when used in conjunction with future Function Block programs, provides custom "outer" control loop capability.

Torque Servo

The Torque Servo configuration provides torque servo control using only the motor mounted feedback device for commutation. Synchronous input data to the servo loop includes only the Torque Offset. This values are updated at the coarse update rate of the associated motion group. The Torque Offset value is derived from the current value of the corresponding attribute. This offset attribute may be changed programmatically via SSV instructions or direct Tag access which, when used in conjunction with future Function Block programs, provides custom "outer" control loop capability.

Drive Gains

Rockwell Automation servo drives use Nested Digital Servo Control Loop such as shown in the block diagrams above, consisting typically of a position loop with proportional, integral and feed-forward gains around a digitally synthesized inner velocity loop, again with proportional and integral gains for each axis. These gains provide software control over the servo dynamics, and allow the servo system to be completely stabilized. Unlike analog servo controllers, these digitally set gains do not drift. Furthermore, once these gains are set for a particular system, another SERCOS module programmed with these gain values will operate identically to the original one.

Notes:

Axis Data Types

Introduction

When you add an axis to your project, RSLogix 5000 software makes a tag for the axis. The tag stores status and fault information for the axis. The layout of the tag depends on the type of axis.

For This Type of Axis	See Page
AXIS_CONSUMED	399
AXIS_GENERIC	402
AXIS_SERVO	404
AXIS_SERVO_DRIVE	407
AXIS_VIRTUAL	411

AXIS_CONSUMED

Member	Data Type	Style
AxisFault	DINT	Hex
PhysicalAxisFault	BOOL	Decimal
ModuleFault	BOOL	Decimal
ConfigFault	BOOL	Decimal
AxisStatus	DINT	Hex
ServoActionStatus	BOOL	Decimal
DriveEnableStatus	BOOL	Decimal
ShutdownStatus	BOOL	Decimal
ConfigUpdateInProcess	BOOL	Decimal
InhibitStatus	BOOL	Decimal
MotionStatus	DINT	Hex
AccelStatus	BOOL	Decimal
DecelStatus	BOOL	Decimal
MoveStatus	BOOL	Decimal
JogStatus	BOOL	Decimal
GearingStatus	BOOL	Decimal
HomingStatus	BOOL	Decimal
StoppingStatus	BOOL	Decimal
AxisHomedStatus	BOOL	Decimal

Member	Data Type	Style
PositionCamStatus	BOOL	Decimal
TimeCamStatus	BOOL	Decimal
PositionCamPendingStatus	BOOL	Decimal
TimeCamPendingStatus	BOOL	Decimal
GearingLockStatus	BOOL	Decimal
PositionCamLockStatus	BOOL	Decimal
MasterOffsetMoveStatus	BOOL	Decimal
CoordinatedMotionStatus	BOOL	Decimal
AxisEvent	DINT	Hex
WatchEventArmedStatus	BOOL	Decimal
WatchEventStatus	BOOL	Decimal
RegEvent1ArmedStatus	BOOL	Decimal
RegEvent1Status	BOOL	Decimal
RegEvent2ArmedStatus	BOOL	Decimal
RegEvent2Status	BOOL	Decimal
HomeEventArmedStatus	BOOL	Decimal
HomeEventStatus	BOOL	Decimal
OutputCamStatus	DINT	Hex
OutputCamPendingStatus	DINT	Hex
OutputCamLockStatus	DINT	Hex
OutputCamTransitionStatus	DINT	Hex
ActualPosition	REAL	Float
StrobeActualPosition	REAL	Float
StartActualPosition	REAL	Float
AverageVelocity	REAL	Float
ActualVelocity	REAL	Float
ActualAcceleration	REAL	Float
WatchPosition	REAL	Float
Registration1Position	REAL	Float
Registration2Position	REAL	Float
Registration1Time	DINT	Decimal
Registration2Time	DINT	Decimal
InterpolationTime	DINT	Decimal
InterpolatedActualPosition	REAL	Float
MasterOffset	REAL	Float
StrobeMasterOffset	REAL	Float
StartMasterOffset	REAL	Float

Member	Data Type	Style
CommandPosition	REAL	Float
StrobeCommandPosition	REAL	Float
StartCommandPosition	REAL	Float
CommandVelocity	REAL	Float
CommandAcceleration	REAL	Float
InterpolatedCommandPosition	REAL	Float
ModuleFaults	DINT	Hex
ControlSyncFault	BOOL	Decimal

AXIS_GENERIC

Member	Data Type	Style
AxisFault	DINT	Hex
PhysicalAxisFault	BOOL	Decimal
ModuleFault	BOOL	Decimal
ConfigFault	BOOL	Decimal
AxisStatus	DINT	Hex
ServoActionStatus	BOOL	Decimal
DriveEnableStatus	BOOL	Decimal
ShutdownStatus	BOOL	Decimal
ConfigUpdateInProcess	BOOL	Decimal
InhibitStatus	BOOL	Decimal
MotionStatus	DINT	Hex
AccelStatus	BOOL	Decimal
DecelStatus	BOOL	Decimal
MoveStatus	BOOL	Decimal
JogStatus	BOOL	Decimal
GearingStatus	BOOL	Decimal
HomingStatus	BOOL	Decimal
StoppingStatus	BOOL	Decimal
AxisHomedStatus	BOOL	Decimal
PositionCamStatus	BOOL	Decimal
TimeCamStatus	BOOL	Decimal
PositionCamPendingStatus	BOOL	Decimal
TimeCamPendingStatus	BOOL	Decimal
GearingLockStatus	BOOL	Decimal
PositionCamLockStatus	BOOL	Decimal
MasterOffsetMoveStatus	BOOL	Decimal
CoordinatedMotionStatus	BOOL	Decimal
AxisEvent	DINT	Hex
WatchEventArmedStatus	BOOL	Decimal
WatchEventStatus	BOOL	Decimal
RegEvent1ArmedStatus	BOOL	Decimal
RegEvent1Status	BOOL	Decimal
RegEvent2ArmedStatus	BOOL	Decimal
RegEvent2Status	BOOL	Decimal
HomeEventArmedStatus	BOOL	Decimal

Member	Data Type	Style
HomeEventStatus	BOOL	Decimal
OutputCamStatus	DINT	Hex
OutputCamPendingStatus	DINT	Hex
OutputCamLockStatus	DINT	Hex
OutputCamTransitionStatus	DINT	Hex
ActualPosition	REAL	Float
StrobeActualPosition	REAL	Float
StartActualPosition	REAL	Float
AverageVelocity	REAL	Float
ActualVelocity	REAL	Float
ActualAcceleration	REAL	Float
WatchPosition	REAL	Float
Registration1Position	REAL	Float
Registration2Position	REAL	Float
Registration1Time	DINT	Decimal
Registration2Time	DINT	Decimal
InterpolationTime	DINT	Decimal
InterpolatedActualPosition	REAL	Float
MasterOffset	REAL	Float
StrobeMasterOffset	REAL	Float
StartMasterOffset	REAL	Float
CommandPosition	REAL	Float
StrobeCommandPosition	REAL	Float
StartCommandPosition	REAL	Float
CommandVelocity	REAL	Float
CommandAcceleration	REAL	Float
InterpolatedCommandPosition	REAL	Float

AXIS_SERVO

Member	Data Type	Style
AxisFault	DINT	Hex
PhysicalAxisFault	BOOL	Decimal
ModuleFault	BOOL	Decimal
ConfigFault	BOOL	Decimal
AxisStatus	DINT	Hex
ServoActionStatus	BOOL	Decimal
DriveEnableStatus	BOOL	Decimal
ShutdownStatus	BOOL	Decimal
ConfigUpdateInProcess	BOOL	Decimal
InhibitStatus	BOOL	Decimal
MotionStatus	DINT	Hex
AccelStatus	BOOL	Decimal
DecelStatus	BOOL	Decimal
MoveStatus	BOOL	Decimal
JogStatus	BOOL	Decimal
GearingStatus	BOOL	Decimal
HomingStatus	BOOL	Decimal
StoppingStatus	BOOL	Decimal
AxisHomedStatus	BOOL	Decimal
PositionCamStatus	BOOL	Decimal
TimeCamStatus	BOOL	Decimal
PositionCamPendingStatus	BOOL	Decimal
TimeCamPendingStatus	BOOL	Decimal
GearingLockStatus	BOOL	Decimal
PositionCamLockStatus	BOOL	Decimal
MasterOffsetMoveStatus	BOOL	Decimal
CoordinatedMotionStatus	BOOL	Decimal
AxisEvent	DINT	Hex
WatchEventArmedStatus	BOOL	Decimal
WatchEventStatus	BOOL	Decimal
RegEvent1ArmedStatus	BOOL	Decimal
RegEvent1Status	BOOL	Decimal
RegEvent2ArmedStatus	BOOL	Decimal
RegEvent2Status	BOOL	Decimal
HomeEventArmedStatus	BOOL	Decimal

Member	Data Type	Style
HomeEventStatus	BOOL	Decimal
OutputCamStatus	DINT	Hex
OutputCamPendingStatus	DINT	Hex
OutputCamLockStatus	DINT	Hex
OutputCamTransitionStatus	DINT	Hex
ActualPosition	REAL	Float
StrobeActualPosition	REAL	Float
StartActualPosition	REAL	Float
AverageVelocity	REAL	Float
ActualVelocity	REAL	Float
ActualAcceleration	REAL	Float
WatchPosition	REAL	Float
Registration1Position	REAL	Float
Registration2Position	REAL	Float
Registration1Time	DINT	Decimal
Registration2Time	DINT	Decimal
InterpolationTime	DINT	Decimal
InterpolatedActualPosition	REAL	Float
MasterOffset	REAL	Float
StrobeMasterOffset	REAL	Float
StartMasterOffset	REAL	Float
CommandPosition	REAL	Float
StrobeCommandPosition	REAL	Float
StartCommandPosition	REAL	Float
CommandVelocity	REAL	Float
CommandAcceleration	REAL	Float
InterpolatedCommandPosition	REAL	Float
ServoStatus	DINT	Hex
ProcessStatus	BOOL	Decimal
OutputLimitStatus	BOOL	Decimal
PositionLockStatus	BOOL	Decimal
HomeInputStatus	BOOL	Decimal
Reg1InputStatus	BOOL	Decimal
Reg2InputStatus	BOOL	Decimal
DriveFaultInputStatus	BOOL	Decimal
ServoFault	DINT	Hex
PosSoftOvertravelFault	BOOL	Decimal

Member	Data Type	Style
NegSoftOvertravelFault	BOOL	Decimal
FeedbackFault	BOOL	Decimal
FeedbackNoiseFault	BOOL	Decimal
PositionErrorFault	BOOL	Decimal
DriveFault	BOOL	Decimal
ModuleFaults	DINT	Hex
ControlSyncFault	BOOL	Decimal
ModuleSyncFault	BOOL	Decimal
TimerEventFault	BOOL	Decimal
ModuleHardwareFault	BOOL	Decimal
InterModuleSyncFault	BOOL	Decimal
AttributeErrorCode	INT	Hex
AttributeErrorID	INT	Hex
PositionCommand	REAL	Float
PositionFeedback	REAL	Float
AuxPositionFeedback	REAL	Float
PositionError	REAL	Float
PositionIntegratorError	REAL	Float
VelocityCommand	REAL	Float
VelocityFeedback	REAL	Float
VelocityError	REAL	Float
VelocityIntegratorError	REAL	Float
AccelerationCommand	REAL	Float
AccelerationFeedback	REAL	Float
ServoOutputLevel	REAL	Float
MarkerDistance	REAL	Float
VelocityOffset	REAL	Float
TorqueOffset	REAL	Float

AXIS_SERVO_DRIVE

Member	Data Type	Style
AxisFault	DINT	Hex
PhysicalAxisFault	BOOL	Decimal
ModuleFault	BOOL	Decimal
ConfigFault	BOOL	Decimal
AxisStatus	DINT	Hex
ServoActionStatus	BOOL	Decimal
DriveEnableStatus	BOOL	Decimal
ShutdownStatus	BOOL	Decimal
ConfigUpdateInProcess	BOOL	Decimal
InhibitStatus	BOOL	Decimal
MotionStatus	DINT	Hex
AccelStatus	BOOL	Decimal
DecelStatus	BOOL	Decimal
MoveStatus	BOOL	Decimal
JogStatus	BOOL	Decimal
GearingStatus	BOOL	Decimal
HomingStatus	BOOL	Decimal
StoppingStatus	BOOL	Decimal
AxisHomedStatus	BOOL	Decimal
PositionCamStatus	BOOL	Decimal
TimeCamStatus	BOOL	Decimal
PositionCamPendingStatus	BOOL	Decimal
TimeCamPendingStatus	BOOL	Decimal
GearingLockStatus	BOOL	Decimal
PositionCamLockStatus	BOOL	Decimal
MasterOffsetMoveStatus	BOOL	Decimal
CoordinatedMotionStatus	BOOL	Decimal
AxisEvent	DINT	Hex
WatchEventArmedStatus	BOOL	Decimal
WatchEventStatus	BOOL	Decimal
RegEvent1ArmedStatus	BOOL	Decimal
RegEvent1Status	BOOL	Decimal
RegEvent2ArmedStatus	BOOL	Decimal
RegEvent2Status	BOOL	Decimal
HomeEventArmedStatus	BOOL	Decimal

Member	Data Type	Style
HomeEventStatus	BOOL	Decimal
OutputCamStatus	DINT	Hex
OutputCamPendingStatus	DINT	Hex
OutputCamLockStatus	DINT	Hex
OutputCamTransitionStatus	DINT	Hex
ActualPosition	REAL	Float
StrobeActualPosition	REAL	Float
StartActualPosition	REAL	Float
AverageVelocity	REAL	Float
ActualVelocity	REAL	Float
ActualAcceleration	REAL	Float
WatchPosition	REAL	Float
Registration1Position	REAL	Float
Registration2Position	REAL	Float
Registration1Time	DINT	Decimal
Registration2Time	DINT	Decimal
InterpolationTime	DINT	Decimal
InterpolatedActualPosition	REAL	Float
MasterOffset	REAL	Float
StrobeMasterOffset	REAL	Float
StartMasterOffset	REAL	Float
CommandPosition	REAL	Float
StrobeCommandPosition	REAL	Float
StartCommandPosition	REAL	Float
CommandVelocity	REAL	Float
CommandAcceleration	REAL	Float
InterpolatedCommandPosition	REAL	Float
ModuleFaults	DINT	Hex
ControlSyncFault	BOOL	Decimal
ModuleSyncFault	BOOL	Decimal
TimerEventFault	BOOL	Decimal
ModuleHardwareFault	BOOL	Decimal
SERCOSRingFault	BOOL	Decimal
AttributeErrorCode	INT	Hex
AttributeErrorID	INT	Hex
PositionCommand	REAL	Float
PositionFeedback	REAL	Float

Member	Data Type	Style
AuxPositionFeedback	REAL	Float
PositionError	REAL	Float
PositionIntegratorError	REAL	Float
VelocityCommand	REAL	Float
VelocityFeedback	REAL	Float
VelocityError	REAL	Float
VelocityIntegratorError	REAL	Float
AccelerationCommand	REAL	Float
AccelerationFeedback	REAL	Float
MarkerDistance	REAL	Float
VelocityOffset	REAL	Float
TorqueOffset	REAL	Float
TorqueCommand	REAL	Float
TorqueFeedback	REAL	Float
PosDynamicTorqueLimit	REAL	Float
NegDynamicTorqueLimit	REAL	Float
MotorCapacity	REAL	Float
DriveCapacity	REAL	Float
PowerCapacity	REAL	Float
BusRegulatorCapacity	REAL	Float
MotorElectricalAngle	REAL	Float
TorqueLimitSource	DINT	Hex
DCBusVoltage	DINT	Decimal
DriveStatus	DINT	Hex
ProcessStatus	BOOL	Decimal
BusReadyStatus	BOOL	Decimal
HomeInputStatus	BOOL	Decimal
Reg1InputStatus	BOOL	Decimal
Reg2InputStatus	BOOL	Decimal
PosOvertravelInputStatus	BOOL	Decimal
NegOvertravelInputStatus	BOOL	Decimal
EnableInputStatus	BOOL	Decimal
AccelLimitStatus	BOOL	Decimal
AbsoluteReferenceStatus	BOOL	Decimal
VelocityLockStatus	BOOL	Decimal
VelocityStandstillStatus	BOOL	Decimal
VelocityThresholdStatus	BOOL	Decimal

Member	Data Type	Style
TorqueThresholdStatus	BOOL	Decimal
TorqueLimitStatus	BOOL	Decimal
VelocityLimitStatus	BOOL	Decimal
PositionLockStatus	BOOL	Decimal
PowerLimitStatus	BOOL	Decimal
LowVelocityThresholdStatus	BOOL	Decimal
HighVelocityThresholdStatus	BOOL	Decimal
DriveFault	DINT	Hex
PosSoftOvertravelFault	BOOL	Decimal
NegSoftOvertravelFault	BOOL	Decimal
PosHardOvertravelFault	BOOL	Decimal
NegHardOvertravelFault	BOOL	Decimal
MotFeedbackFault	BOOL	Decimal
MotFeedbackNoiseFault	BOOL	Decimal
AuxFeedbackFault	BOOL	Decimal
AuxFeedbackNoiseFault	BOOL	Decimal
DriveEnableInputFault	BOOL	Decimal
CommonBusFault	BOOL	Decimal
PreChargeOverloadFault	BOOL	Decimal
GroundShortFault	BOOL	Decimal
DriveHardFault	BOOL	Decimal
OverSpeedFault	BOOL	Decimal
OverloadFault	BOOL	Decimal
DriveOvertempFault	BOOL	Decimal
MotorOvertempFault	BOOL	Decimal
DriveCoolingFault	BOOL	Decimal
DriveControlVoltageFault	BOOL	Decimal
FeedbackFault	BOOL	Decimal
CommutationFault	BOOL	Decimal
DriveOvercurrentFault	BOOL	Decimal
DriveOvervoltageFault	BOOL	Decimal
DriveUndervoltageFault	BOOL	Decimal
PowerPhaseLossFault	BOOL	Decimal
PositionErrorFault	BOOL	Decimal
SERCOSFault	BOOL	Decimal
SERCOSErrorCode	INT	Hex

AXIS_VIRTUAL

Member	Data Type	Style
AxisFault	DINT	Hex
PhysicalAxisFault	BOOL	Decimal
ModuleFault	BOOL	Decimal
ConfigFault	BOOL	Decimal
AxisStatus	DINT	Hex
ServoActionStatus	BOOL	Decimal
DriveEnableStatus	BOOL	Decimal
ShutdownStatus	BOOL	Decimal
ConfigUpdateInProcess	BOOL	Decimal
InhibitStatus	BOOL	Decimal
MotionStatus	DINT	Hex
AccelStatus	BOOL	Decimal
DecelStatus	BOOL	Decimal
MoveStatus	BOOL	Decimal
JogStatus	BOOL	Decimal
GearingStatus	BOOL	Decimal
HomingStatus	BOOL	Decimal
StoppingStatus	BOOL	Decimal
AxisHomedStatus	BOOL	Decimal
PositionCamStatus	BOOL	Decimal
TimeCamStatus	BOOL	Decimal
PositionCamPendingStatus	BOOL	Decimal
TimeCamPendingStatus	BOOL	Decimal
GearingLockStatus	BOOL	Decimal
PositionCamLockStatus	BOOL	Decimal
MasterOffsetMoveStatus	BOOL	Decimal
CoordinatedMotionStatus	BOOL	Decimal
AxisEvent	DINT	Hex
WatchEventArmedStatus	BOOL	Decimal
WatchEventStatus	BOOL	Decimal
RegEvent1ArmedStatus	BOOL	Decimal
RegEvent1Status	BOOL	Decimal
RegEvent2ArmedStatus	BOOL	Decimal
RegEvent2Status	BOOL	Decimal
HomeEventArmedStatus	BOOL	Decimal

Member	Data Type	Style
HomeEventStatus	BOOL	Decimal
OutputCamStatus	DINT	Hex
OutputCamPendingStatus	DINT	Hex
OutputCamLockStatus	DINT	Hex
OutputCamTransitionStatus	DINT	Hex
ActualPosition	REAL	Float
StrobeActualPosition	REAL	Float
StartActualPosition	REAL	Float
AverageVelocity	REAL	Float
ActualVelocity	REAL	Float
ActualAcceleration	REAL	Float
WatchPosition	REAL	Float
Registration1Position	REAL	Float
Registration2Position	REAL	Float
Registration1Time	DINT	Decimal
Registration2Time	DINT	Decimal
InterpolationTime	DINT	Decimal
InterpolatedActualPosition	REAL	Float
MasterOffset	REAL	Float
StrobeMasterOffset	REAL	Float
StartMasterOffset	REAL	Float
CommandPosition	REAL	Float
StrobeCommandPosition	REAL	Float
StartCommandPosition	REAL	Float
CommandVelocity	REAL	Float
CommandAcceleration	REAL	Float
InterpolatedCommandPosition	REAL	Float

Coordinate System Attributes

Use that attributes of a coordinate system for information about the coordinate system.

How to Access Attributes

The Access column shows how to access the attribute

Example

Attribute	Axis Type	Data Type	Access	Description
Actual Position Tolerance			GSV SSV	
Config Fault			Tag	
Coordinate Motion Status			GSV Tag	

Use a Get System Value (GSV) instruction to get the value.

Use a Set System Value (SSV) instruction to set or change the value.

Use the tag for the coordinate system to get the value.

Use the tag for the coordinate system or a GSV instruction to get the value. It's easier to use the tag.

Coordinate System Attributes

Attribute	Data Type	Access	Description
Accel Status	BOOL	Tag	Use the Accel Status bit to determine if the coordinated (vectored) motion is currently being commanded to accelerate. The acceleration bit is set when a coordinated move is in the accelerating phase due to the current coordinated move. It is cleared when the coordinated move has been stopped or the coordinated move is in the decelerating phase.
Actual Pos Tolerance Status	BOOL	Tag	Use the Actual Pos Tolerance Status bit to determine when a coordinate move is within the Actual Position Tolerance. The Actual Position Tolerance Status bit is set for AT term type only. The bit is set when interpolation is complete and the actual distance to programmed endpoint is less than the configured AT value. The bit remains set after an instruction completes. The bit is reset if either a new instruction is started or the axis moves such that the actual distance to programmed endpoint is greater than the configured AT value
Actual Position	REAL[8]	Tag	Array of actual position of each axis associated to this motion coordinate system in Coordinate Units.
Actual Position Tolerance	REAL	GSV SSV	Coordination Units The Actual Position Tolerance attribute value is a distance unit used when instructions such as MCLM, MCCM and so on specify a Termination Type of Actual Position.
Axes Configuration Faulted	DINT	GSV Tag	Shows which axes in this coordinate system have a configuration fault. **If this bit is on** / **Then this axis has a configuration fault** 0 / 0 1 / 1 2 / 2
Axes Inhibited Status	DINT	GSV Tag	Shows which axes in this coordinate system are inhibited. **If this bit is on** / **Then this axis is inhibited** 0 / 0 1 / 1 2 / 2
Axes Servo On Status	DINT	GSV Tag	Shows which axes in this coordinate system are on (via MSO). **If this bit is on** / **Then this axis is on** 0 / 0 1 / 1 2 / 2

<table>
<tr><th>Attribute</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Axes Shutdown Status</td><td>DINT</td><td>GSV

Tag</td><td>Shows which axes in this coordinate system are shutdown.
<table><tr><th>If this bit is on</th><th>Then this axis is shutdown</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr><tr><td>2</td><td>2</td></tr></table></td></tr>
<tr><td>Axis Fault</td><td>DINT</td><td>GSV

Tag</td><td>The Axis Fault Bits attribute is a roll-up of all of the axes associated to this motion coordinate system. A bit being set indicates that one of the associated axes has that fault.
<table><tr><th>Type</th><th>Bit</th></tr><tr><td>Physical Axis Fault</td><td>0</td></tr><tr><td>Module Fault</td><td>1</td></tr><tr><td>Config Fault</td><td>2</td></tr></table></td></tr>
<tr><td>Axis Inhibit Status</td><td>BOOL</td><td>Tag</td><td>If this bit is:
· ON — An axis in the coordinate system is inhibited.
· OFF — None of the axis in the coordinate system are inhibited.</td></tr>
<tr><td>Command Pos Tolerance Status</td><td>BOOL</td><td>Tag</td><td>Use the Command Position Tolerance Status bit to determine when a coordinate move is within the Command Position Tolerance.

The Command Position Tolerance Status bit is set for all term types whenever the distance to programmed endpoint is less than the configured CT value. The bit will remains set after an instruction completes. The bit is reset when a new instruction is started.</td></tr>
<tr><td>Command Position Tolerance</td><td>REAL</td><td>GSV

SSV</td><td>Coordination Units

The Command Position Tolerance attribute value is a distance unit used when instructions such as MCLM, MCCM and so on specify a Termination Type of Command Position.</td></tr>
<tr><td>Config Fault</td><td>BOOL</td><td>Tag</td><td>The Configuration Fault bit is set when an update operation targeting an axis configuration attribute of an associated motion module has failed. Specific information concerning the Configuration Fault may be found in the Attribute Error Code and Attribute Error ID attributes associated with the motion module.</td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Coordinate Motion Status</td><td>DINT</td><td>GSV

Tag</td><td>Lets you access the motion status bits for the coordinate system in one 32-bit word.

<table><tr><th>Status</th><th>Bit</th></tr><tr><td>Accel Status</td><td>0</td></tr><tr><td>Decel Status</td><td>1</td></tr><tr><td>Actual Pos Tolerance Status</td><td>2</td></tr><tr><td>Command Pos Tolerance Status</td><td>3</td></tr><tr><td>Stopping Status</td><td>4</td></tr><tr><td>Reserved</td><td>5</td></tr><tr><td>Move Status</td><td>6</td></tr><tr><td>Transition Status</td><td>7</td></tr><tr><td>Move Pending Status</td><td>8</td></tr><tr><td>Move Pending Queue Full Status</td><td>9</td></tr></table></td></tr>
<tr><td>Coordinate System Auto Tag Update</td><td>SINT</td><td>GSV

SSV</td><td>The Coordinate System Auto Tag Update attribute configures whether the Actual Position attribute is automatically updated each motion task scan. This is similar to, but separate from the Motion Group's "Auto Tag Update" attribute.

0 – auto update disabled

1 – auto update enabled (default)</td></tr>
<tr><td>Coordinate System Status</td><td>DINT</td><td>GSV

Tag</td><td>Lets you access the status bits for the coordinate system in one 32-bit word.

<table><tr><th>Status</th><th>Bit</th></tr><tr><td>Shutdown Status</td><td>0</td></tr><tr><td>Ready Status</td><td>1</td></tr><tr><td>MotionStatus</td><td>2</td></tr><tr><td>Axis Inhibit Status</td><td>3</td></tr></table></td></tr>
<tr><td>Decel Status</td><td>BOOL</td><td>Tag</td><td>Use the Decel Status bit to determine if the coordinated (vectored) motion is currently being commanded to decelerate.

The deceleration bit is set when a coordinated move is in the decelerating phase due to the current coordinated move. It is cleared when the coordinated move has been stopped or the coordinated move is complete.</td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Dynamics Configuration Bits</td><td>DINT</td><td>GSV
SSV</td><td>Revision 16 improved how the controller handles changes to an S-curve profile.

Do you want to return to revision 15 or earlier behavior for S-curves?

· NO — Leave these bits ON (default).
· YES — Turn OFF one or more of these bits.

<table><tr><th>To turn off this change</th><th>Turn off this bit</th></tr><tr><td>Reduced S-curve Stop Delay

This change applies to the Motion Coordinated Stop (MCS) instruction. It lets you use a higher deceleration jerk to stop an accelerating coordinate system more quickly.

The controller uses the deceleration jerk of the stopping instruction if it is more than the current acceleration jerk.</td><td>0</td></tr><tr><td>Reduced S-curve Velocity Reversals

Before revision 16, you could cause a coordinate system to momentarily reverse direction if you decreased the deceleration jerk while the coordinate system was decelerating. This typically happened if you tried to restart a move with a lower deceleration rate while the coordinate system was stopping. This change prevents the coordinate system from reversing in those situations.</td><td>1</td></tr><tr><td>Reduced S-curve Velocity Overshoots

You can cause a coordinate system to overshoot its programmed speed if you decrease the acceleration jerk while the coordinate</td><td>2</td></tr></table></td></tr>
<tr><td>Maximum Acceleration</td><td>REAL</td><td>GSV

SSV</td><td>Coordination Units / Sec^2

The Maximum Acceleration attribute value is used by motion instructions such as MCLM, MCCM and so on, to determine the acceleration rate to apply to the coordinate system vector when the acceleration is specified as a percent of the Maximum.</td></tr>
<tr><td>Maximum Deceleration</td><td>REAL</td><td>GSV

SSV</td><td>Coordination Units / Sec^2

The Maximum Deceleration attribute value is used by motion instructions such as MCLM, MCCM and so on, to determine the deceleration rate to apply to the coordinate system vector when the deceleration is specified as a percent of the Maximum.</td></tr>
<tr><td>Maximum Pending Moves</td><td>DINT</td><td>GSV</td><td>The Maximum Pending Moves attribute is used to determine how many Move Pending queue slots should be created as part of the Coordinate System's create service.

Limited to a queue of one.</td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Maximum Speed</td><td>REAL</td><td>GSV

SSV</td><td>Coordination Units / Sec

The value of the Maximum Speed attribute is used by various motion instructions (for example, MCLM, MCCM and so on) to determine the steady-state speed of the coordinate system vector when the speed is specified as a percent of the Maximum.</td></tr>
<tr><td>Module Fault</td><td>BOOL</td><td>Tag</td><td>The Module Fault bit attribute is set when a serious fault has occurred with the motion module associated with the selected axis. Usually a module fault affects all axes associated with the motion module. A module fault generally results in the shutdown of all associated axes. Reconfiguration of the motion module is required to recover from a module fault condition.</td></tr>
<tr><td>Modules Faulted</td><td>DINT</td><td>GSV

Tag</td><td>Shows which axes in this coordinate system have a module fault.

<table><tr><th>If this bit is on</th><th>Then this axis has a module fault</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr><tr><td>2</td><td>2</td></tr></table></td></tr>
<tr><td>Motion Status</td><td>BOOL</td><td>Tag</td><td>The Motion Status bit attribute is set indicating that at least one Coordinate Motion instruction is active and the Coordinate System is connected to its associated axes.</td></tr>
<tr><td>Move Pending Queue Full Status</td><td>BOOL</td><td>Tag</td><td>The move pending queue full bit is set there is no room in the instruction queue for the next coordinated move instruction. Once there is room in the queue, the bit is cleared.</td></tr>
<tr><td>Move Pending Status</td><td>BOOL</td><td>Tag</td><td>The move pending bit is set once a coordinated motion instruction is queued. Once the instruction has begun executing, the bit will be cleared, provided no subsequent coordinated motion instructions have been queued in the mean time. In the case of a single coordinated motion instruction, the status bit may not be detected by the user in RSLogix5000 since the transition from queued to executing is faster than the coarse update. The real value of the bit comes in the case of multiple instructions. As long as an instruction is in the instruction queue, the pending bit will be set. This provides the RSLogix5000 programmer a means of stream-lining the execution of multiple coordinated motion instructions. Ladder logic containing coordinated motion instructions can be made to execute faster when the programmer allows instructions to be queued while a preceding instruction is executing. When the MovePendingStatus bit is clear, the next coordinated motion instruction can be executed (that is, setup in the queue).</td></tr>
<tr><td>Move Status</td><td>BOOL</td><td>Tag</td><td>The move bit is set when coordinated motion is generating motion for any associated axes. Once coordinated motion is no longer being commanded, the move bit is cleared.</td></tr>
<tr><td>Move Transition Status</td><td>BOOL</td><td>Tag</td><td>The move transition bit is set once the blend point between two successive coordinated moves has been reach. The bit remains set while the blend of the two moves into one is in process. Once the blend is complete, the move transition bit is cleared.</td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Data Type</th><th>Access</th><th>Description</th></tr>
<tr><td>Physical Axes Faulted</td><td>DINT</td><td>GSV
Tag</td><td>Shows which axes in this coordinate system have a servo axis fault.
<table>
<tr><th>If this bit is on</th><th>Then this axis has a servo axis fault</th></tr>
<tr><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td></tr>
<tr><td>2</td><td>2</td></tr>
</table></td></tr>
<tr><td>Physical Axis Fault</td><td>BOOL</td><td>Tag</td><td>If the Physical Axis Fault bit is set, it indicates that there is one or more fault conditions have been reported by the physical axis. The specific fault conditions can then be determined through access to the fault attributes of the associated physical axis.</td></tr>
<tr><td>Ready Status</td><td>BOOL</td><td>Tag</td><td>The Ready bit is set when all associated axes are enabled. It is cleared after an MCSD, MGSD or a fault on any of the associated axes.</td></tr>
<tr><td>Shutdown Status</td><td>BOOL</td><td>Tag</td><td>The Coordinate System bit will be set after an MCSD or MGSD is executed and all associated axes have stopped. A MCSR or a MGSR will reset the coordinate system and clear the bit. Coordinated moves cannot be initiated while this bit is set.</td></tr>
<tr><td>Stopping Status</td><td>BOOL</td><td>Tag</td><td>The stopping bit is set when a MCS instruction is executed. The bit will remain set until all coordinated motion is stopped. The bit is cleared when all coordinated motion has stopped.</td></tr>
<tr><td>Transform Source Status</td><td>BOOL</td><td>Tag</td><td>If the bit is:
· ON — The coordinate system is the source of an active transform.
· OFF — The coordinate system isn't the source of an active transform.</td></tr>
<tr><td>Transform Target Status</td><td>BOOL</td><td>Tag</td><td>If the bit is:
· ON — The coordinate system is the target of an active transform.
· OFF — The coordinate system isn't the target of an active transform.</td></tr>
</table>

Notes:

Symbols

89

Numerics

1394C Drive module
inhibit an axis 72
1394-CFLAExx Cable
Pinouts 264
Wiring Diagram 264
1398-CFLAExx
Cable Diagram 260
Pinouts 260
1756-HYD02
add to controller 15
1756-M02AE
add to controller 15
1756-M02AE servo module
Block diagrams
Torque servo drive 388
Velocity servo drive 389
Features 9
Loop and interconnect diagrams 387
Troubleshooting 133
Wiring diagrams
1394 drive 263
Servo module RTB 258
Ultra 100 drive 259
Ultra 200 drive 259
Ultra3000 drive 261
1756-M02AS
add to controller 15
1756-M03SE
add to controller 15
set up 17
1756-M08SE
add to controller 15
set up 17
1756-M16SE
add to controller 15
set up 17

A

Arm Solution
definition of
configuring 127
Articulated Dependent
base offsets 99
configuring 92
define configuration parameters 98
end effector offsets 100
establish the reference frame 92
establish the reference frame alternate methods 94
identify the work envelope 96
link lengths 99
Articulated Independent
base offsets 89
configuration parameters 88
end effector offsets 90
establish reference frame 82, 88
establish reference frame methods 84
identify the work envelope 87
link lengths 89
axis
add to controller 20
check wiring 25
get status 30
inhibit 69, 75
set up 22
tune 26
Axis Properties
Aux Feedback Tab - AXIS_SERVO_DRIVE 185
Aux Feedback Tab (AXIS_SERVO_DRIVE)
Cycles 185
Feedback Ratio 186
Feedback Type 185
Interpolation Factor 185
Per 185
Conversion Tab 187
Conversion Constant 188
Position Unwind 188
Positioning Mode 187
Drive/Motor Tab - (AXIS_SERVO_DRIVE) 177
Amplifier Catalog Number 177
Attribute 1/Atrribute 2 179
Calculate button 181
Calculate Parameters 183
Per 182
Position Range 182
Position Unit Scaling 182
Position Unit Unwind 182
Change Catalog Button 180
Catalog Number 181
Filters 181
Family 181
Feedback Type 181
Voltage 181
Drive Enable Input Checking 179
Drive Enable Input Fault 179
Drive Resolution 179
Loop Configuration 178
Real Time Axis Information 179

Drive/Motor Tab (AXIS_SERVO_DRIVE)
(Motor) Catalog Number 177
Dynamics Tab 205
Calculate
Maximum Acceleration Jerk
Maximum Deceleration Jerk 209
Manual Tune 208, 209
Maximum Acceleration 206
Maximum Acceleration Jerk 207
Maximum Deceleration 206
Maximum Deceleration Jerk 208
Maximum Velocity 206
Fault Actions Tab - AXIS_SERVO 246
Drive Fault 248
Feedback Loss 248
Feedback Noise 248
Position Error 248
Soft Overtravel 249
Fault Actions Tab - AXIS_SERVO_DRIVE 249
Drive Thermal 251
Feedback 252
Feedback Noise 251
Hard Overtravel 252
Motor Thermal 251
Position Error 252
Set Custom Stop Action 253
Soft Overtravel 252
Feedback Tab - AXIS_SERVO 172
Feedback Type 172
A Quadrature B Encoder Interface (AQB 172
Linear Displacement Transducer (LDT) 173
Absolute Feedback Offset 176
Calculated Values 176
Calculate Button 176
Conversion Constant 176
Minimum Servo Update Period 176
Calibration Constant 175
Enable Absolute Feedback 176
LDT Type 175
Length 175
Recirculations 175
Scaling 176
Synchronous Serial Interface (SSI) 172
Absolute Feedback Offset 174
Clock Frequency 174
Code Type 173
Data Length 173
Enable Absolute Feedback 174
Gains Tab - AXIS_SERVO
Differential 213
Integral (Position) Gain 212
Integrator Hold 214
Manual Tune 215
Proportional (Position) Gain 212
Proportional (Velocity) Gain 213
Gains Tab - AXIS_SERVO_DRIVE 210, 215
Acceleration Feedforward 214, 217
Integral (Position) Gain 218
Integral (Velocity) Gain 213, 219
Integrator Hold 220
Manual Tune 220
Proportional (Position) Gain 218
Proportional (Velocity) Gain 213, 219
Set Custom Gains 221
Velocity Feedforward 214, 217
Homing Tab - AXIS_VIRTUAL 193
Mode 197
Position 197
Sequence 198
Homing Tab - SERVO_AXIS and SERVO_AXIS_DRIVE 188
Direction 191, 196
Limit Switch 191, 195
Mode 189, 193
Offset 191, 195
Position 190, 194
Return Speed 192, 196
Sequence 191, 195
Speed 192, 196
Hookup Tab - AXIS_SERVO 198
Feedback Polarity 198
Output Polarity 199
Test Feedback 199
Test Increment 198
Test Marker 199
Test Output & Feedback 199
Hookup Tab Overview - AXIS_SERVO_DRIVE 200
Drive Polarity 200
Test Feedback 201
Test Increment 200
Test Marker 201
Test Output & Feedback 201

Limits Tab - AXIS_SERVO 229
Manual Tune 232
Maximum Negative 231
Maximum Positive 231
Output Limit 232
Position Error Tolerance 231
Soft Travel Limits 231
Limits Tab - AXIS_SERVO_DRIVE 233
Continuous Torque/Force Limit 235
Hard Travel Limits 234
Manual Tune 236
Maximum Negative 234
Maximum Positive 234
Peak Torque/Force Limit 235
Position Error Tolerance 234
Position Lock Tolerance 235
Set Custom Limits 236
Soft Travel Limits 234
Motor/Feedback Tab (AXIS_SERVO_DRIVE) 184
(Motor) Cycles 184
(Motor) Feedback Type 184
(Motor) Interpolation Factor 184
Per 184
Offset Tab - AXIS_SERVO 239
Backlash Compensation 241
Reversal Offset 241
Stabilization Window 241
Friction/Deadband Compensation 240
Friction Compensation 240
Friction Compensation Window 240
Manual Tune 242
Output Offset 241
Torque Offset 241
Velocity Offset 241
Offset Tab - AXIS_SERVO_DRIVE 242
Backlash Compensation 244
Reversal Offset 244
Stabilization Window 245
Friction Compensation 243
Friction Compensation Window 244
Manual Tune 245
Torque Offset 245
Velocity Offset 245
Output Tab - SERVO_AXIS 222
Enable Low-pass Output Filter 224
Low-pass Output Filter Bandwidth 224
Manual Tune 225
Torque Scaling 223
Velocity Scaling 223
Output Tab Overview - AXIS_SERVO_DRIVE 225
Enable Low-pass Output Filter 228
Enable Notch Filter 227
Load Inertia Ratio 227
Low-pass Output Filter Bandwidth 228
Manual Tune 229
Motor Inertia 227
Notch Filter 227
Torque Scaling 227
Servo Tab - AXIS_SERVO 170
Direct Drive Ramp Rate 171
Drive Fault Input 171
Enable Direct Drive Ramp Control 171
Enable Drive Fault Input 171
External Drive Configuration 170
Hydraulic 170
Torque 170
Velocity 170
Loop Configuration 170
Real Time Axis Information 171
Attribute 1/Attribute 2 171
Tag Tab 254
Data Type 256
Description 255
Name 255
Scope 256
Style 256
Tag Type 255
Tune Tab - AXIS_SERVO, AXIS_SERVO_DRIVE 202
Damping Factor 203
Direction 203
Speed 202
Start Tuning 204
Torque (AXIS_SERVO) 203
Torque/Force (AXIS_SERVO_DRIVE) 202
Travel Limit 202
Tune 204
AXIS Structures 273
AXIS_ CONSUMED 273
AXIS_SERVO 273
AXIS_SERVO_DRIVE 273
AXIS_VIRTUAL 273
Axis Tag types
alias tag 48
base tag 48

B

Base Offsets
definition of 89
Block diagrams for a 1756-M02AE module 387
With a torque servo drive 388
With a velocity servo drive 389

C

Cartesian Gantry
configuration parameters 101
configure 101
establish reference frame 101
identify the work envelope 101
Cartesian H-bot
base offsets 104
configuration parameters 104
configure 102
end effector offsets 104
establish reference frame 103
identify the work envelope 104
link lengths 104
Catalog 181
coarse update period
set 18
Configure 108, 111
configure
SERCOS interface module 17
coordinate system
overview 30
Coordinate System Properties
Dynamics Tab 62
Manual Adjust 65
Reset Button 66
Manual Adjust Button 65
Position Tolerance Box 64
Actual 65
Command 65
Vector Box 62
Maximum Acceleration 63
Maximum Deceleration 63
Maximum Speed 63
Editing 51
General Tab 53
Axis Grid 55
Axis Name 55
Coordinate 55
Coordination Mode 55
Ellipsis Button (...) 55
Dimension 54
Ellipsis button 54
Enable Coordinate System Auto Tag Update 56
Motion Group 54
New Group button 54
Transform Dimension 54
Type 54
Geometry Tab 57
Link Lengths 57
zero angle orientations box 57
Joints Tab 61
Joint Ratio 61
Joint Units 62
Offsets Tab 60
Base Offsets 60
Tag Tab 66
Data Type 67
Description 67
Name 66
Scope 67
Tag Type 67
Units Tab
Axis Grid 58
Axis Name 59
Conversion Ratio 59
Conversion Ratio Units 59
Coordination Units 58
Coordinate system properties
Offsets Tab
End Effector 60
coordinated system time master
set 14
CST master
See coordinated system time master

D

Delta Robot
Maximum Negative Joint Limit Condition 114
Maximum Positive Joint Limit Condition 114, 115, 117, 118, 119, 120, 122, 123, 124
types
configure 108
Delta three-dimensional
configuration parameters 115
configure 108
maximum positive joint limit condition 114
reference frame 110
work envelope 112
zero angle orientation 111
Delta two-dimensional
configuration parameters 120

configure 117
establish the reference frame 118
work envelope 119
Diagrams
Block 387
diagrams
wiring 257
Direct Commands
Accessing
From Group 32
Supported Commands
Motion State 34
drive
add SERCOS interface drive 16
check wiring 25

E

Editing Axis Properties
General Tab – AXIS_GENERIC 165
Axis Configuration 165
Channel 166
Ellipsis (…) button 166
Module 166
Motion Group 165
General Tab - AXIS_SERVO_DRIVE 160, 164
Assigned Motion Group 161
Axis Configuration 160
Module 161
Module Type 161
Node 161
Node with a Kinetix 6000 Drive 162
General Tab – SERVO_AXIS 159
Axis Configuration 159
Channel 160
Module 160
Motion Planner Tab 166
Enable Master Position Filter Checkbox 168
Master Delay Compensation Checkbox 167
Master Position Filter Bandwidth 168
Output Cam Execution Targets 166
Program Stop Action 167
Units Tab 169
Average Velocity Timebase 169
Position Units 169
Encoder 298
Encoder:Noise 282, 321, 337
End Effector Offsets
determining 99
Establish 110

G

General 50
General Tab - AXIS_VIRTUAL 164
Assigned Motion Group 164
Geometry
of robot 80
tab 51
Geometry Tab
link lengths 57
zero angle orientations box 57

H

home limit switch
wire diagram 271
home limit switch input wire 271
hookup tests
run 25

I

Identify 112
inhibit
axis 69, 75
axis of a 1394 drive 72

K

Kinematics
activating 128
arm solutions 127, 130
arm solutions for two axes robots 127
Articulated Independent 82
changing arm solutions 129
determine Coordinate system type 80
error conditions 131
no solution 130
overview 75
singularity 130
solution mirroring 127
terms 77

L

Linear displacement transducer (LDT)
Connecting the LDT to the 1756-HYD02 module ??-266, 268-??

M

Motion Apply Axis Tuning 35

Motion Apply Hookup Diagnostic 35
Motion Arm Output Cam 35
Motion Arm Registration 35
Motion Arm Watch Position 35
Motion Attributes
Axis Event Bit Attributes 288
Axis Fault Bit Attributes 288
Axis Status Bit Attributes 291
Commissioning Configuration Attributes
Damping Factor 298
Drive Model Time Constant 304
Position Servo Bandwidth 352
Test Increment 366
Tuning Configuration Bits 374
Bi-directional Tuning 375
Tune Acceleration Feedforward 374
Tune Friction Compensation 375
Tune Output Low-Pass Filter 374
Tune Position Error Integrator 374
Tune Torque Offset 375
Tune Velocity Error Integrator 374
Tune Velocity Feedforward 374
Tuning Direction Reverse 374
Tuning Speed 375
Tuning Torque 375
Tuning Travel Limit 376
Velocity Servo Bandwidth 383
Configuration Attributes
Axis Type 292
Motion Conversion Configuration
Conversion Constant 298
Motion Dynamics Configuration
Maximum Acceleration 333
Maximum Deceleration 333
Maximum Speed 334
Programmed Stop Mode 354
Fast Disable 354
Fast Shutdown 354
Fast Stop 354
Hard Disable 354
Hard Shutdown 354
Motion Homing Configuration
Active Homing
Active Immediate Home 154
Home Configuration Bits 325
Home Switch Normally Closed 325
Home Mode 326
Home Offset 326
Home Position 327
Home Return Speed 327, 328
Home Sequence and Home Direction 325, 327
Home Speed 327
Passive Homing
Passive Home with Marker 157
Passive Home with Switch 157
Passive Home with Switch then Marker 157
Passive Immediate Home 157
Motion Planner Configuration Attributes
Master Input Configuration Bits 331, 332
Master Delay Compensation 331
Master Position Filter 332
Master Position Filter Bandwidth 332
Output Cam Execution Targets 343
Motion Unit Configuration Attributes
Average Velocity Timebase 285
Position Units 352
Position Unwind 352
Rotary Axis 358
Interface Attributes
Axis Configuration State 285
Axis Data Type 287
Consumed 287
Feedback 287
Generic 287
Servo 287
Servo Drive 287
Virtual 287
Axis Instance 290
Axis State 291
C2C Connection Instance 295
C2C Map Instance 295
Group Instance 325
Home Event Task Instance 326
Map Instance 330
Memory Usage 334
Memory Use 334
Module Channel 334

Module Class Code 335
Registration 1 Event Task Instance 355
Registration 2 Event Task Instance 355
Watch Event Task Instance 384
Module Fault Bit Attribute 335
Motion Coordinate System
Status Attributes
Axis Fault 415
Faulted 414, 418, 419
Servo On Axes 414
Shutdown 415
Coordinate Motion Status 413, 416
Coordinate System Status 416
Motion Coordinate System Configuration Attributes
Coordinate System Auto Tag Update 416
Coordinate System Dynamics Configuration
Actual Position Tolerance 413, 414
Command Position Tolerance 415
Maximum Acceleratio 417
Maximum Deceleration 417
Maximum Speed 418
Max Pending Moves 417
Motion Status Attributes
Actual Acceleration 273, 279
Actual Position 279
Actual Velocity 279
Average Velocity 284
Command Acceleration 295
Command Position 296
Command Velocity 296
Interpolated Actual Position 328
Interpolated Command Position 328
Interpolation Time 328
Master Offset 332
Motion Status Bits 338
Registration Position 356
Registration Time 356
Start Master Offset 364
Start Position 364
Strobe Master Offset 365
Strobe Position 365
Watch Position 384
Servo Configuration Attributes
Absolute Feedback Enable 274
Absolute Feedback Offset 275
Axis Info Select 289
External Drive Type 317
Fault Configuration Bits 318
Drive Fault Checking 318
Drive Fault Normally Closed 319
Hard Overtravel Checking 318
Soft Overtravel Checking 318
LDT Calibration Constant 329
LDT Calibration Constant Units 329
LDT Length 329
LDT Length Units 329
LDT Recirculations 329
LDT Scaling 329
LDT Scaling Units 329
LDT Type 329
Servo Feedback Type 360
A Quadrature B Encoder Interface 360
Linear Displacement Transducer 361
Synchronous Serial Interfac 360
Servo Loop Configuration 361
Servo Polarity Bits 362
Feedback Polarity Negative 362
Servo Polarity Negative 362
SSI Clock Frequency 363
SSI Code Type 363
SSI Data Length 363
Servo Drive Attributes
Analog Input 280
Attribute Error Code 280
Attribute Error ID 280
Axis Control Bit Attributes 286
Abort Process 286
Change Cmd Reference 286
Shutdown Request 286
Axis Info Select 289
Axis Response Bit Attributes 290
Abort Event Acknowledge 290
Abort Home Acknowledge 290
Abort Process Acknowledge 290
Change Pos Reference 290
Shutdown Request Acknowledge 290
Commissioning Configuration Attributes
Motor Inertia & Load Inertia Ratio 330, 341
Commissioning Status Attributes
Test Direction Forward 366
Test Status 366

Tune Acceleration 371
Tune Acceleration Time 371
Tune Deceleration 371
Tune Deceleration Time 371
Tune Inertia 372
Tune Status 373
Drive Fault Bit Attributes 303
Drive Gains 397
Advanced Drive Gain Attributes 376
Output Notch Filter Frequency 345
Velocity Proportional Gain Maximum Bandwidth 381
Drive Limits
Advanced Drive Limits 278, 279, 368, 370, 379, 383, 384
Continuous Torque Limit 297
Torque Limit 368
Drive Offsets
Backlash Reversal Error 292
Backlash Stabilization Window 293
Drive Fault Actions 300, 313, 341
Advanced Stop Action Attributes 364, 365
Brake Engage Delay 294
Brake Release Delay 294
Resistive Brake Contact Delay 357
Drive Power Attributes
Bus Regulator ID 295
Power Supply ID 353
PWM Frequency Select 354
Drive Warning Bit Attributes 315
Cooling Error Warning 315
Drive Overtemperature Warning 315
Motor Overtemperature Warning 315
Overload Warning 315
Module Fault Bit Attributes 335
Module Hardware Fault 335, 336
Timer Event Fault 335, 336
Motor and Feedback Configuration
Aux Feedback Ratio 282
Feedback Configuration 281, 339
Feedback Polarity 281, 339
Feedback Interpolation 282, 339
Feedback Resolution 283, 340
Feedback Type 283, 340
Feedback Units 283, 340
Motor Data 338
Motor ID 341
SERCOS Error Code 358
Servo Drive Configuration Attributes
Advanced Scaling Attributes 309
Data Reference 310
Linear Scaling Unit 310
Scaling Type 309
Scaling Unit 309
Advanced Servo Configuration Attributes 353, 365
Drive ID 299
Drive Polarity 305
Advanced Polarity Attributes 350, 369, 379
Custom Polarity 305
Negative Polarity 305
Positive Polarity 305
Drive Resolution 306
Drive Travel Range Limit 306
Drive Units 314
Fault Configuration Bits 318
Drive Enable Input Checking 319
Drive Enable Input Fault Handling 319
Hard Overtravel Checking 318
Soft Overtravel Checkin 318
Fractional Unwind 306
Linear Ball-Screw WITHOUT Aux Feedback Device 307
Linear Ball-Screw/Ball-Screw Combination WITH Aux Feedback Device 308
Rotary Gear-Head WITH Aux Feedback Device 307
Rotary Gear-Head WITHOUT Aux Feedback Device 307
Servo Loop Configuration 361
Servo Loop Block Diagrams 390
Auxiliary Dual Command Servo 395

Auxiliary Position Servo 392
Dual Command Feedback Servo 395
Dual Feedback Servo 393
Motor Dual Command Servo 394
Motor Position Servo 391
Torque Servo 396
Velocity Servo 396
Servo Drive Status Attributes
Acceleration Command 276
Acceleration Feedback 276
Aux Position Feedback 284
Bus Regulator Capacity 294, 295
DC Bus Voltage 299
Drive Capacity 299
Drive Status Bit Attributes 311
Marker Distance 330
Motor Capacity 338
Motor Electrical Degrees 338
Negative Dynamic Torque Limi 342
Position Command 347
Position Error 348
Position Feedback 349
Position Integrator Error 350
Positive Dynamic Torque Limit 353
Power Capacity 353
Torque Command 367
Torque Feedback 367
Torque Limit Source 369
Velocity Command 376
Velocity Error 376
Velocity Feedback 377
Velocity Integrator Error 379
Servo Fault Configuration
Servo Fault Actions 302, 320, 321, 325, 348, 363
Servo Gains
Acceleration Feedforward Gain 273, 277, 413
Bandwidth Method 351
Integrator Hold Enable 328
Loop Gain Method 351
Maximum Bandwidth 351
Position Differential Gain 347
Position Integral Gain 349
Position Proportional Gain 351
Velocity Feedforward Gain 377
Velocity Integral Gain 378
Velocity Proportional Gain 380
Backlash Reversal Error 292
Backlash Stabilization Window 293
Directional Scaling Ratio 299
Maximum Bandwidth 380
Output LP Filter Bandwidth 345
Torque Scaling 370
Velocity Scaling 382
Servo Limits
Direct Drive Ramp Rate 299
Friction Compensation 321
Friction Compensation Window 322
Maximum Negative Travel 333
Maximum Positive Travel 334
Output Limit 344
Output Offset 345
Position Error Tolerance 348
Position Lock Tolerance 350
Torque Offset 369
Velocity Offset 379
Servo Loop Block Diagrams 388
Position Servo with Torque Servo Drive 388
Position Servo with Velocity Servo Drive 389
Servo Status Attributes
Acceleration Command 276
Acceleration Feedback 276
Attribute Error Code 280
Attribute Error ID 280
Aux Position Feedback 284
Axis Response Bit Attributes
Zero DAC Request Acknowledge 290
Commissioning Status Attributes
Test Direction Forward 366
Test Status 366
Tune Acceleration 371
Tune Acceleration Time 371
Tune Deceleration 371
Tune Deceleration Time 371
Tune Inertia 372
Tune Rise Time 373
Tune Speed Scaling 373
Tune Status 373
Marker Distance 330
Position Command 347
Position Error 348
Position Feedback 349
Position Integrator Error 350
Servo Fault Bit Attributes 359
Servo Output Level 362
Servo Status Bit Attributes 363
Velocity Command 376
Velocity Error 376

Velocity Feedbac 377
Velocity Integrator Error 379
Status Attributes
Output Cam Lock Status 343
Output Cam Pending Status 343
Output Cam Status 343
Output Cam Transition Status 344
Motion Axis Fault Reset 34
Motion Axis Gear 34
Motion Axis Home 34
Motion Axis Jog 34
Motion Axis Move 34
Motion Axis Position Cam 34
Motion Axis Shutdown 34
Motion Axis Shutdown Reset 34
Motion Axis Stop 34
Motion Axis Time Cam 34
Motion Calculate Cam Profile 34
Motion Calculate Slave Values 34
Motion Change Dynamics 34
motion control
add axis 20
choose a motion module 15
coarse update period 18
coordinate system 30
execution 18
overview 13
program 28
set the coordinated system time master 14
set up an axis 22
status information 30
Motion Coordinated Change Dynamics 36
Motion Coordinated Circular Move 36
Motion Coordinated Linear Move 36
Motion Coordinated Shutdown 36
Motion Coordinated Shutdown Reset 36
Motion Coordinated Stop 36
Motion Direct Commands 31
Error Process 39
Transition States 42
Motion Direct Drive Off 34
Motion Direct Drive On 34
Motion Disarm Output Cam 35
Motion Disarm Registration 35
Motion Disarm Watch Position 35
motion group
set up 18
Motion Group Shutdown 35
Motion Group Shutdown Reset 35
Motion Group Stop 35
Motion Group Strobe Position 35
Motion Instructions 31
Coordinated Motion Instructions
Motion Coordinated Change Dynamics (MCCD) 36
Motion Coordinated Circular Move (MCCM) 36
Motion Coordinated Linear Move (MCLM) 36
Motion Coordinated Shutdown (MCSD) 36
Motion Coordinated Shutdown Reset (MCSR) 36
Motion Coordinated Stop (MCS) 36
Motion Configuration Instructions
Motion Apply Axis Tuning (MAAT) 35
Motion Apply Hookup Diagnostic (MAHD) 35
Motion Run Axis Tuning (MRAT) 35
Motion Run Hookup Diagnostic (MRHD) 35
Motion Direct Commands 31
Motion Event Instructions
Motion Arm Output Cam (MAOC) 35
Motion Arm Registration (MAR) 35
Motion Arm Watch Position (MAW) 35
Motion Disarm Output Cam (MDOC) 35
Motion Disarm Registration (MDR) 35
Motion Disarm Watch Position (MDW) 35
Motion Group Instructions
Motion Group Shutdown (MGSD) 35
Motion Group Shutdown Reset (MGSR) 35
Motion Group Stop (MGS) 35
Motion Group Strobe Position (MGSP) 35
Motion Move Instructions
Motion Axis Gear (MAG) 34
Motion Axis Home (MAH) 34
Motion Axis Jog (MAJ) 34
Motion Axis Move (MAM) 34
Motion Axis Position Cam (MAPC) 34
Motion Axis Stop (MAS) 34
Motion Axis Time Cam (MATC) 34
Motion Calculate Cam Profile (MCCP) 34
Motion Calculate Slave Values 34

Motion Change Dynamics (MCD) 34
Motion Redefine Position (MRP) 34
Motion State Instructions
Motion Axis Fault Reset (MAFR) 34
Motion Axis Shutdown (MASD) 34
Motion Axis Shutdown Reset (MASR) 34
Motion Direct Drive Off (MDF) 34
Motion Direct Drive On (MDO) 34
Motion Servo Off (MSF) 34
Motion Servo On (MSO) 34
motion instructions
overview 28
motion planner
set period 18
Motion Redefine Position 34
Motion Run Axis Tuning 35
Motion Run Hookup Diagnostic 35
Motion Servo Off 34
Motion Servo On 34

N

Naming a Coordinate System 48
Entering Tag Information 48
Parameters 49
Alias For 49
Data Type 49
Description 49
Name 49
Scope 49
Style 49
Tag Type 49
Alias 49
Base 49

O

Offsets 51
OK contact wire 271
OK contacts
wire diagram 271

R

registration sensor
wiring diagram 270
RSLogix 5000 programming software
Motion Instructions 31

S

SCARA
configure 104
SCARA Delta
configuation parameters 124
establish the reference frame 122
identify the work envelope 123
SCARA Independent
configure 104
reference frame 104, 106
Selective Compliant Assembly Robot Arm
base offsets 108
configuration parameters 107
configure 104, 127
end effector offsets 108
establish reference frame 104
identify work envelope 106
link lengths 107
SERCOS interface drive
add to controller 16
SERCOS interface module
choose 15
set up 17
Singularity
planning for
definition of 130
Specifications 9
1756-HYD02 Motion Module 9
1756-M02AE Motion Module 9
1756-M02AS Motion Module 9
1756-M03SE, 1756-M08SE, & 1756-M16SE Motion Module 9
structures
AXIS 273

T

Troubleshooting 133
1756-HYD02 Module LED 138
DRIVE Indicator 140
1756-M02AE LED 133
DRIVE LED indicator 134
1756-M02AS LED 135
FDBK Indicator 136
1756-M08SE LED
SERCOS interface LED 141
1756-M16SE LED
SERCOS interface LED 141
SERCOS interface LED Indicators 141
tune
axis 26

U

Units 51

W

Wiring connections ??-266
Connecting LDTs to the 1756-HYD02 module ??-266, 268-??
Example diagram of 1756-HYD02 wiring 267
wiring connections
home limit switch input 271
OK contacts 271
Wiring diagrams
1394 drive 263
registration sensor 270
Servo module RTB 258
Ultra 100 drive 259
Ultra 200 drive 259
Ultra3000 Drive 261
wiring diagrams 257
home limit switch 271
OK contacts 271
Wizard
dynamics 51
general 50
geometry 51
offset 51
tag 51
units 51

How Are We Doing?

Your comments on our technical publications will help us serve you better in the future. Thank you for taking the time to provide us feedback.

You can complete this form and mail (or fax) it back to us or email us at RADocumentComments@ra.rockwell.com.

Pub. Title/Type Motion Modules in Logix5000 Control Systems

Cat. No. *1756-HYD02, 1756-L60M03SE, 1756-M02AE, 1756-M02AS, 1756-M03SE, 1756-M08SE, 1756-M16SE, 1768-M04SE* Pub. No. LOGIX-UM002D-EN-P Pub. Date July 2008 Part No. 953030-71

Please complete the sections below. Where applicable, rank the feature (1=needs improvement, 2=satisfactory, and 3=outstanding).

Overall Usefulness 1 2 3	How can we make this publication more useful for you?	
Completeness (all necessary information is provided) 1 2 3	Can we add more information to help you?	
	procedure/step	illustration feature
	example	guideline other
	explanation	definition
Technical Accuracy (all provided information is correct) 1 2 3	Can we be more accurate?	
	text	illustration
Clarity (all provided information is easy to understand) 1 2 3	How can we make things clearer?	
Other Comments	You can add additional comments on the back of this form.	

Your Name _____

Your Title/Function _____

Location/Phone _____

Would you like us to contact you regarding your comments?

___No, there is no need to contact me

___Yes, please call me

___Yes, please email me at _____

___Yes, please contact me via _____

Return this form to: Rockwell Automation Technical Communications, 1 Allen-Bradley Dr., Mayfield Hts., OH 44124-9705

Fax: 440-646-3525 Email: RADocumentComments@ra.rockwell.com

Publication CIG-CO521D-EN-P- July 2007

PLEASE FASTEN HERE (DO NOT STAPLE)

Other Comments

PLEASE FOLD HERE

PLEASE REMOVE

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 18235 CLEVELAND OH

POSTAGE WILL BE PAID BY THE ADDRESSEE

Rockwell Automation

1 ALLEN-BRADLEY DR
MAYFIELD HEIGHTS OH 44124-9705

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe/Middle East/Africa: Rockwell Automation, Vorstlaan/Boulevard du Souverain 36, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication LOGIX-UM002D-EN-P - July 2008
Supersedes Publication LOGIX-UM002C-EN-P - July 2007

Copyright © 2008 Rockwell Automation, Inc. All rights reserved. Printed in the U.S.A.

Rockwell Automation Support

Rockwell Automation provides technical information on the Web to assist you in using its products. At http://support.rockwellautomation.com, you can find technical manuals, a knowledge base of FAQs, technical and application notes, sample code and links to software service packs, and a MySupport feature that you can customize to make the best use of these tools.

For an additional level of technical phone support for installation, configuration, and troubleshooting, we offer TechConnect support programs. For more information, contact your local distributor or Rockwell Automation representative, or visit http://support.rockwellautomation.com.

Installation Assistance

If you experience a problem within the first 24 hours of installation, please review the information that's contained in this manual. You can also contact a special Customer Support number for initial help in getting your product up and running.

United States	1.440.646.3434 Monday – Friday, 8am – 5pm EST
Outside United States	Please contact your local Rockwell Automation representative for any technical support issues.

New Product Satisfaction Return

Rockwell Automation tests all of its products to ensure that they are fully operational when shipped from the manufacturing facility. However, if your product is not functioning and needs to be returned, follow these procedures.

United States	Contact your distributor. You must provide a Customer Support case number (call the phone number above to obtain one) to your distributor in order to complete the return process.
Outside United States	Please contact your local Rockwell Automation representative for the return procedure.

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe/Middle East/Africa: Rockwell Automation, Vorstlaan/Boulevard du Souverain 36, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication LOGIX-UM002D-EN-P - July 2008
Supersedes Publication LOGIX-UM002C-EN-P - July 2007

Copyright © 2008 Rockwell Automation, Inc. All rights reserved. Printed in the U.S.A.